History Matching In Parallel Computational Environments

Annual Report

Reporting period start date: 1 Oct. 2003
Reporting period end date: 30 Sept. 2004

Authors: Dr. Steven Bryant, Dr. Sanjay Srinivasan, Alvaro Barrera and Sharad Yadav

Date Issued: October 2004

DOE Award Number: DE-FC26-03NT15410

Submitting organization: Center for Petroleum and Geosystems Engineering
The University of Texas at Austin
Austin, Texas 78712-0228

DISCLAIMER

ABSTRACT

In the probabilistic approach for history matching, the information from the dynamic data is merged with the prior geologic information in order to generate permeability models consistent with the observed dynamic data as well as the prior geology. The relationship between dynamic response data and reservoir attributes may vary in different regions of the reservoir due to spatial variations in reservoir attributes, fluid properties, well configuration, flow constrains on wells etc. This implies probabilistic approach should then update different regions of the reservoir in different ways. This necessitates delineation of multiple reservoir domains in order to increase the accuracy of the approach.

The research focuses on a probabilistic approach to integrate dynamic data that ensures consistency between reservoir models developed from one stage to the next. The algorithm relies on efficient parameterization of the dynamic data integration problem and permits rapid assessment of the updated reservoir model at each stage. The report also outlines various domain decomposition schemes from the perspective of increasing the accuracy of probabilistic approach of history matching.

Research progress in three important areas of the project are discussed:

- Validation and testing the probabilistic approach to incorporating production data in reservoir models.
- Development of a robust scheme for identifying reservoir regions that will result in a more robust parameterization of the history matching process.
- Testing commercial simulators for parallel capability and development of a parallel algorithm for history matching.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Validating Probabilistic Approach for Dynamic Data Integration

Reservoir Models are generated considering subsurface geological data obtained from different sources (such as seismic, well logging, well tests, stratigraphy, etc), and a geological model of heterogeneity. The variogram model is commonly used as the heterogeneity model and can be inferred from the same conditional subsurface information. These two components are combined within a simulation/interpolation framework to generate geological models conditioned to static information.

Geostatistics as a geological modeling technique and its two original basis, the variogram model and the kriging interpolation methodology, were initiated by the work of Daniel Krige (1951) and built upon by Georges Matheron (1962-1963, 1965), with the purpose of providing locally accurate grade estimates of mining blocks; however, its application has extended from the mining industry to many other related disciplines, including the oil industry.

The simple kriging (SK) estimator k_{SK}^{*} at each location u_j of the target geological model $k(u)$ (such permeability or porosity field) is the best linear unbiased estimator and can be written as:

$$k_{SK}^{*} - m(u_j) = \sum_{i=1}^{N} \lambda_i(u_j)\left[k(u_i) - m(u_i)\right]$$

where, $m(u_j) = E\{k(u_j)\}$, $j = 1,...,J$, are the known stationary means of the random function $k(u_j)$ at the locations u_j; J is the size of the model; $k(u_i), i = 1,...,N$ are the conditional data; and $\lambda_i(u_j)$ are the kriging weights for each conditional data at each location for the estimation at location u_j. The weights are calculated from the following system of equations:

$$\sum_{k=1}^{N} \lambda_k(u_j) C(u_i - u_k) = C(u_i - u_j), \forall i = 1,...,N$$

where $C(u_i - u_j)$ is the covariance between locations u_i and u_j, and is related to the variogram by: $\gamma(u_i - u_j) = C(0) - C(u_i - u_j)$ and $C(0) = Var\{k(u_{i)}\}$. The corresponding minimum estimation (error) variance σ^2_{SK} is:

$$\sigma^2_{SK}(u_j) = C(0) - \sum_{i=1}^{N} \lambda_i(u_j) C(u_i - u_j)$$

Stochastic simulation was introduced by Matheron (1973) and Journel (1974) to correct for the smoothing effects and other artifacts of kriging (See Figure 1a.), allowing the reproduction of the spatial variance predicted by the variogram model. Different algorithms were developed including sequential simulation (Journel, 1983, Isaaks, 1990; Srivastava, 1992; Goovaerts, 1997; Chiles and Delfiner, 1999), which has become the workhorse for many current geostatistical applications.

The stochastic simulation approach is based on the calculation of probability distributions at individual locations, considering the conditional information and the spatial heterogeneity model. There are different methods for the construction of the local probability distributions, including the Gaussian approach where the kriging estimation and the estimation variance are used as the mean and the variance of the local normal conditional distribution. In another approach, the use of indicator transforms allows modeling multivariate distributions without relying on Gaussian assumption, to generate models that exhibit more connected and well-defined geological bodies. Figure 1 compares the results of the original kriging interpolation technique with the Gaussian and the Indicator Sequential Simulations, considering the same conditional information.

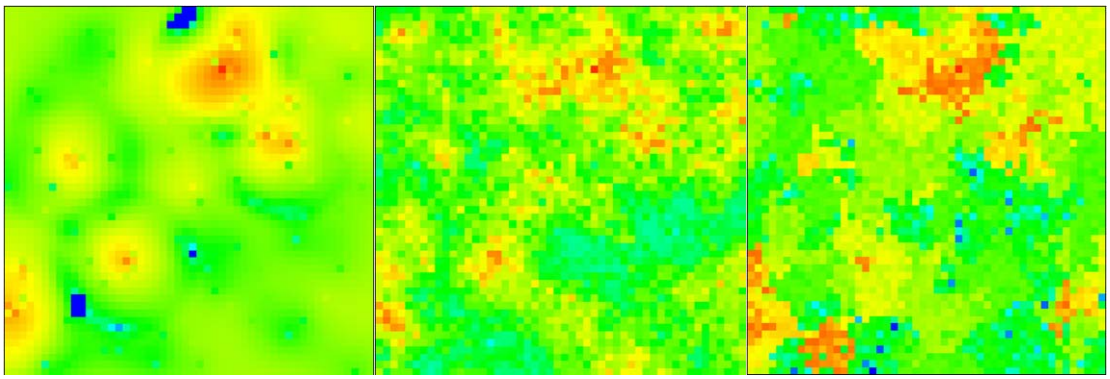


Figure 1: Spatial interpolation obtained by: a) Kriging; b) Sequential Gaussian simulation and c) Sequential indicator simulation.

The sequential simulation paradigm is based on Bayes theorem, where the N-variate probability distribution can be decomposed into the product of N univariate conditional probability distributions. Consequently, the N-variate distribution can be sampled in a sequence of N simulated values drawn from conditional probability distributions

progressively conditioned to more data. Each simulated value is assimilated into the data set and the uncertainty distribution at the next node is thus conditioned to that one additional piece of information. Sequential simulation corrects for the smoothness of the kriged map since the simulated models are samples from the multivariate distribution that depicts the joint variability of all points in space.

Stochastic simulation also provides the capability to generate multiple equiprobable realizations, giving birth to the idea of assessing spatial uncertainty (Journel and Huijbregts, 1978) on reservoir models.

Sequential Indicator Simulation

Spatial distributions can be modeled following a non-parametric approach, where the local probability distributions $F(u; z_i)$ can be calculated from the available conditional information, by defining a set of thresholds $z_i, i = 1, ..., NT$ to discretize the range of variability of the spatial variable, and subsequently performing indicator kriging using the indicator transformed variables.

The indicator transform of a random variable is simply a binary transform: the value one is assigned if the value at a location is less than the threshold and zero if not. The expected value of an indicator random variable is therefore equivalent to the probability of that particular threshold. Hence, the probability distribution can be calculated by sequentially calculating the expected value of the indicator random variable corresponding to different thresholds. An indicator variable is defined as:

$$I(u,z_i)=\begin{cases} i & if\ k(u)\le z_i\ \forall i=1,...,NT \\ NT+1 & if\ k(u)>z_i \end{cases}$$

Hence the indicator transform discretizes a continuous variable (such permeability) into classes or categories. The expected value of the indicator corresponding to a particular category is:

$$E\{I(u,z_i\}=\text{Prob}\{k(u)\le z_i\}=F_k(z_i)$$

The indicator coded data is used to infer the experimental variogram at each threshold, allowing the usage of different heterogeneity models (variograms) for different thresholds. Then, at a particular location, the conditional expectation of the indicator random function for each threshold is determined by applying indicator kriging with the available indicator coded conditional information.

$$I^*\left(u_\alpha;z_i\left|(n)\right.\right)=F_k^*\left(z_i\left|(n)\right.\right)=\sum_{i=1}^{n}\lambda_\alpha(u).i(u_\alpha;z_i)$$

where $i(u_\alpha;z_i)$ is the indicator coded data at location u_α, with n conditional data; and the weights $\lambda_\alpha(u)$ are obtained by solving a kriging system that utilizes indicator covariances:

$$\sum_{\beta=1}^{n}\lambda_\beta(u,z_i)C_I\left(h_{\alpha\beta};z_i\right)=C_I\left(h_{\alpha o};z_i\right),\forall\alpha=1,...,n$$

This conditional expectation is the conditional probability for that particular threshold. The probabilities (conditional expectations) for the local conditional distributions are evaluated at a limited set of thresholds. Therefore, interpolation and extrapolation methods are required to obtain a continuous conditional cumulative distribution function.

Interpolation between the thresholds and tail extrapolations can be obtained applying different approaches such linear or hyperbolic interpolation/extrapolation or using tabulated values.

Following the sequential simulation approach, a realization of the target reservoir model is generated by sequentially sampling from the local conditional distributions following a random path, where the previously sampled values become conditional information for the construction of subsequent local conditional distributions. The process is repeated until all the uninformed locations in the model are populated. Monte Carlo or other sampling technique can be applied on the local conditional distributions. Multiple realizations of the target reservoir can be obtained by altering the random path and/or changing the random draw from the local conditional distributions.

The application of the Indicator sequential simulation approach can be summarized by the following:

1. Select appropriate thresholds consistent with the spatial phenomena.
2. Indicator code the data corresponding to different thresholds
3. Infer indicator variogram/covariance model(s) for different thresholds.
4. Define a random path to visit all uninformed locations. On each subsequent location of the random path apply the following sub-procedure:

4.1. Calculate the conditional expectation of the indicator random function for all the thresholds, applying indicator kriging with the available indicator coded conditional information.

4.2. Correct for order relations (non-monotonicity of the distributions) on evaluated probabilities (conditional expectations).

4.3. Monte Carlo sample a value from the local conditional distribution. In the sampling process, use interpolation/extrapolation methods to model a continuous ccdf from the discrete probabilities evaluated at the thresholds.

4.4. Include the sampled value in the list of conditional information for subsequent estimations.

5. A single realization of the target reservoir is obtained after all the uninformed locations have been visited following the random path. To generate multiple realization repeat step 4 with different random paths.

An indicator sequential simulator has been implemented on C++ language, and validated with other algorithms available on public domain. This algorithm is the base code for the probability updating method utilized in the research project. In subsequent sections, additions including modules for gradual deformation of geological models, interface with flow simulators, and optimization schemes are also developed.

Gradual Deformation of Geological Models using Dynamic Data

Honoring the geological model is an important objective during the generation of static geological models; however, it is commonly forgotten during the integration of dynamic information. During the final stage of reservoir modeling, the history matching process, the perturbations or modification in the model should be conditioned not only to the reproduction of flow history, but also to honor the geological model of heterogeneity. In

the research project this goal is appraised by applying a probabilistic approach for gradual deformation of geological models. The gradual deformation is obtained by systematically perturbing the local conditional distributions with a deformation parameter, rd.

Perturbation of Local Conditional Distributions

Local conditional distributions are perturbed in order to obtain the local distributions conditioned to dynamic information. The deformation parameter, r_d controls the magnitude of perturbation of the conditional distributions. Different perturbation schemes have been evaluated in order to define the methodology that better fits the objectives of the research. The deformation parameter, r_d varies in the range [0, 1].

The gradual deformation starts with a particular realization of the target reservoir, where all the model locations, cells or nodes have a simulated value that falls within a particular indicator category (range between two thresholds, or the upper/lower tails), that is referred as the initial class, z_I. This realization is obtained by following a particular random path, sequentially generating the local conditional distributions and drawing values from them.

During the gradual deformation of the geological model, the random path is fixed. Changing the sampling draws from local conditional distributions will introduce change in the local outcomes, and consequently the deformation in the geological model. The

perturbation in the local conditional distributions is introduced to render the model deformation more gradual, systematic and controlled.

In the first perturbation scheme, the deformation parameter, r_D reduces the probabilities of all indicator categories in the local conditional distribution, except that of the initial class, z_I, which is proportionally increased. In this case the deformation parameter multiplies the probabilities of the off-class indicator categories and the probability of the initial class always increases (or remains the same for $r_d = 1$). This perturbation can be written in terms of the conditional probabilities as:

$$F_k^{'}(z_i \,|(n)) = \begin{cases} r_D \cdot F_k(z_i \,|(n)) & i \neq I \\ 1 - \sum_{\substack{j=1 \\ j \neq I}}^{NT+1} r_D \cdot F_k(z_j \,|(n)) & i = I \end{cases}$$

where $F_k^{'}(z_i \,|(n))$ is the perturbed local conditional probability. A deformation parameter of value zero will generate a distribution with probability 1 for the initial class, z_I, ensuring the reproduction of the initial realization. A deformation parameter of value one will recover the local distribution conditioned to geological information. Since the deformation parameter only increases the probability of the initial class, the maximum deformation of the model is rather small, slowing the process of gradual deformation. This drawback encouraged the evaluation of other perturbation schemes.

In the second perturbation scheme, the deformation parameter reduces the probability of the initial class in the local distributions, while the probabilities of the other indicator categories increase proportionally to their initial values. In this case the probability of the initial class is always reduced (or remains the same for $r_d = 1$). This perturbation scheme can be written as:

$$F_k^{'}(z_i\,|(n)) = \begin{cases} r_D \cdot F_k(z_i\,|(n)) & i = I \\ \dfrac{F_k(z_i\,|(n))\left(1 - r_D \cdot F_k(z_I\,|(n))\right)}{\sum_{\substack{j=1 \\ j \neq I}}^{NT+1} F_k(z_j\,|(n))} & i \neq I \end{cases}$$

A deformation parameter of value zero will generate a distribution with probability 0 for the initial class, producing the maximum deformation of the geological model. A deformation parameter of value one will recover the local distribution conditioned to geological information. Now, the deformation parameter only decreases the probability of the initial class, ensuring the maximum deformation of the geological model, speeding the process of gradual deformation. However, this perturbation scheme does not allow the reproduction of the initial realization and consequently the deformation process is no longer systematic and controlled.

In the third perturbation scheme, the two previous schemes were combined to ensure a more controlled, but at the same time fast gradual deformation of the geological model. In this case the probability of the initial class has a range of variation from 1 to 0 (for r_d = 0 to r_d = 1), ensuring the reproduction of the initial realization for r_d = 0.0, the recovery of the distribution conditioned to geological information for r_d = 0.5, and rejecting the initial class and generating a maximum deformation for r_d = 1.0. In this approach the range of

variation of the deformation parameter, r_d, is divided in two intervals, values below and above 0.5. For values of r_d below 0.5 the first perturbation scheme is applied using a transformed value of r_d, $r_{d'} = 2 \times r_d$. For r_d values above 0.5, the second perturbation scheme is applied with the transformed value of r_d, $r_{d'} = (2 - 2 \times r_d)$. The perturbation scheme can be written as:

$$r_D^{'} = \begin{cases} 2r_D & r_D \le 0.5 \\ 2 - 2r_D & r_D > 0.5 \end{cases}$$

$$F_k^{'}(z_i|(n)) = \begin{cases} r_D^{'} \cdot F_k(z_i|(n)) & i \ne I;\ r_D \le 0.5 \\ 1 - \sum_{\substack{j=1 \\ j \ne I}}^{NT+1} r_D^{'} \cdot F_k(z_j|(n)) & i = I;\ r_D \le 0.5 \\ r_D^{'} \cdot F_k(z_i|(n)) & i = I;\ r_D > 0.5 \\ \dfrac{F_k(z_i|(n))\left(1 - r_D^{'} \cdot F_k(z_I|(n))\right)}{\sum_{\substack{j=1 \\ j \ne I}}^{NT+1} F_k(z_j|(n))} & i \ne I;\ r_D > 0.5 \end{cases}$$

Consequently, the probability of the initial class is increased from the original value for r_d values below 0.5; and decreased for rd values above 0.5, ensuring a wider but controlled range of variation in the perturbation of the local distribution. Figure 2 shows the effect of the deformation parameter r_D on the local conditional distribution under the perturbation scheme 3. Figure 3 shows the effect of the deformation parameter on the geological model.

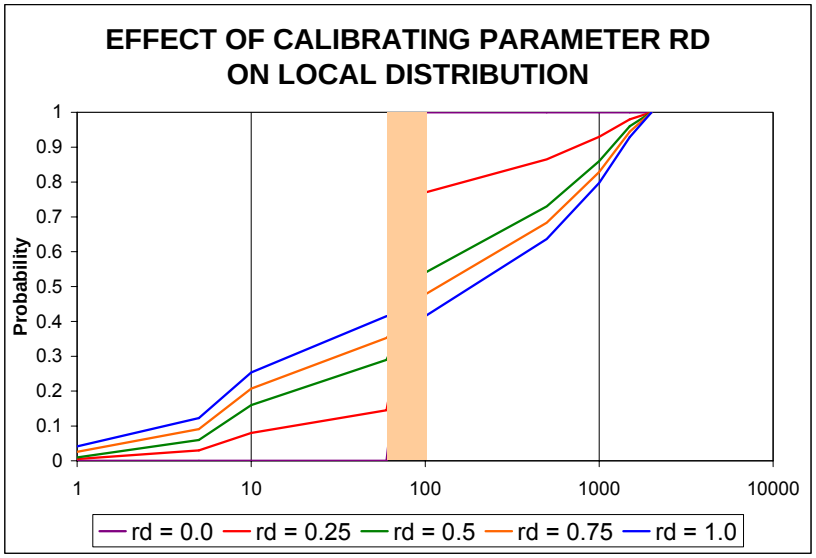


Figure 2: Effect of deformation parameter on local distribution. For rd = 0.5, the local distribution conditioned to geological information is recovered. The initial category, z_I, is represented by the shaded area.

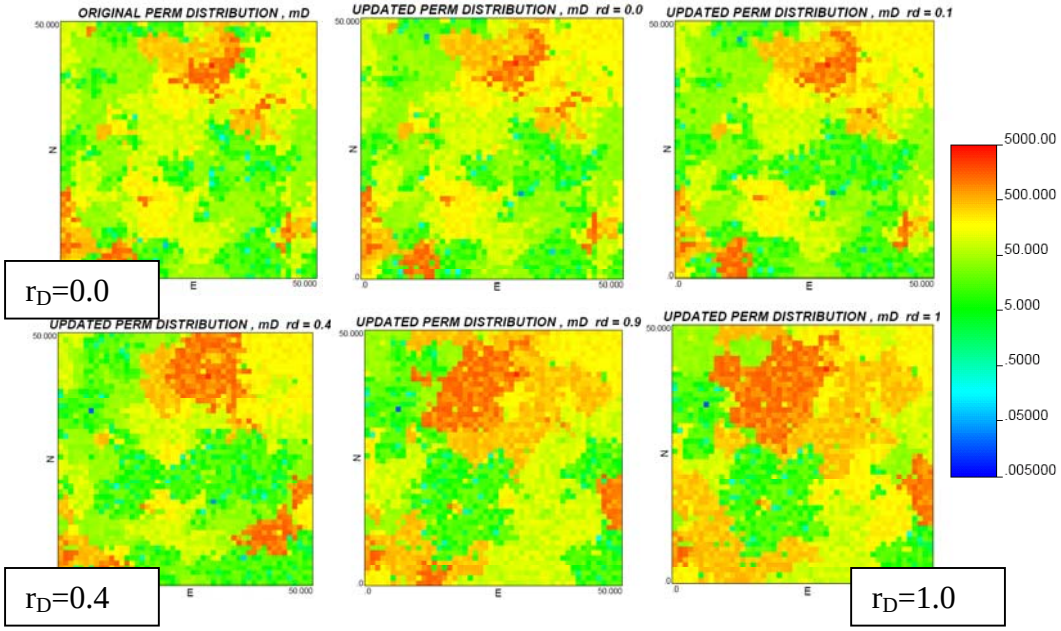


Figure 3. Example of gradual deformation of geological models by probability perturbation method. A single parameter, r_d determines the magnitude of the perturbation in this model.

Yet another perturbation scheme is to be evaluated; a fourth case. In this case the perturbation introduced by the deformation parameter in the local distributions is controlled in order to generate a gradual transition between the local conditional

distribution of two different equiprobable realizations obtained with different random paths and different sampling draws. The deformation parameter plays the role of the weight in the weighted average between the corresponding probabilities of the indicator categories in the local distributions of both realizations. Consequently, values of 0.0 or 1.0 for the deformation parameter will ensure the reproduction of either of the two initial realizations. The motivation behind this fourth perturbation scheme is the gradual deformation of geological models between two previously known states or realizations, differing from the three previous perturbation schemes where only the initial state of the gradual deformation is known.

Conditioning of Local Distribution to Dynamic Information

Now that the methodology for the perturbation of local conditional distributions in geological models has been defined, the next step is to couple the perturbation methodology with the dynamic response of the model. The ultimate goal of estimating the local probability distributions of the geological event A, (permeability, porosity, etc) conditioned to the dynamic information, C, i.e. P(A|C) requires the implementation of an optimization scheme to calibrate the deformation parameter, and the development of interfaces between the geological modeling algorithm and a flow simulator. Figure 4 shows the impact of the deformation parameter on the flow response of the perturbed geological model.

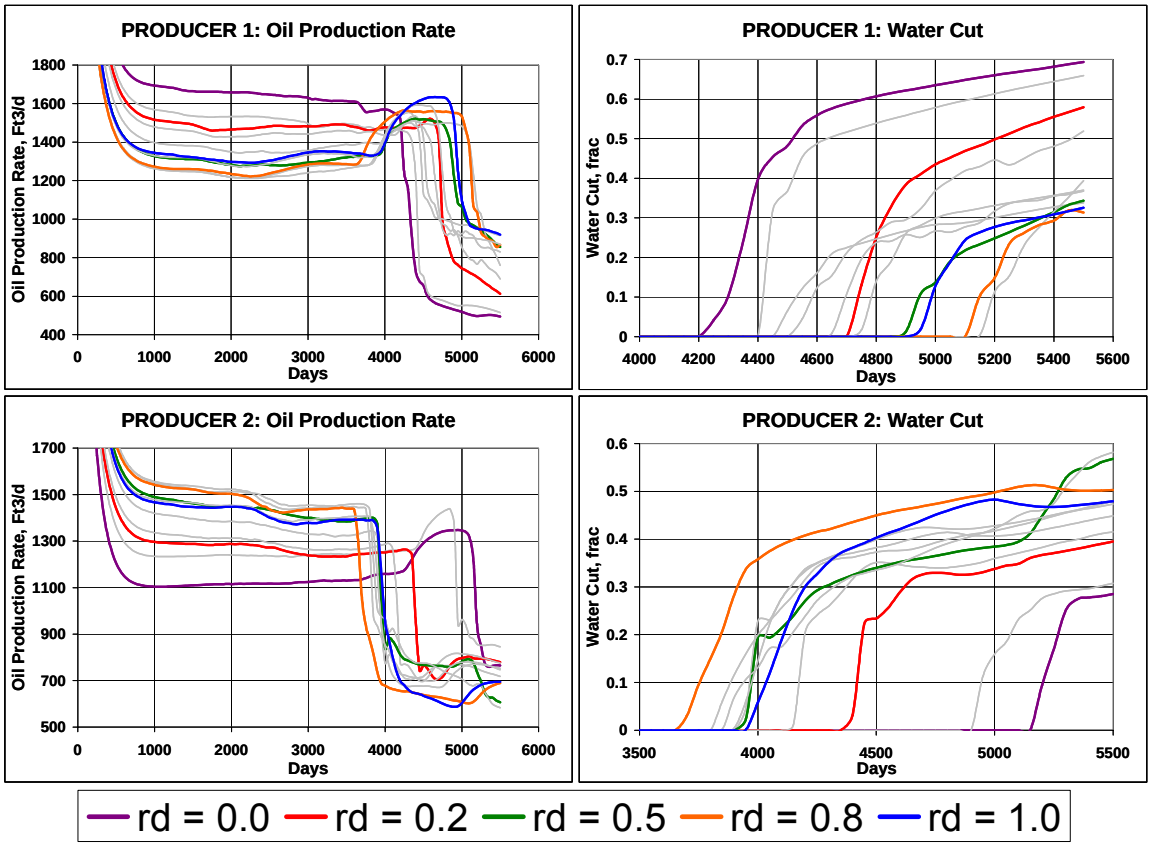


Figure 4: Effect of varying the r_D parameter on: a) Oil production rate at Producer 1; b) Water cut at Producer 1; c) Oil production rate at Producer 2; d) Water cut at Producer 2.

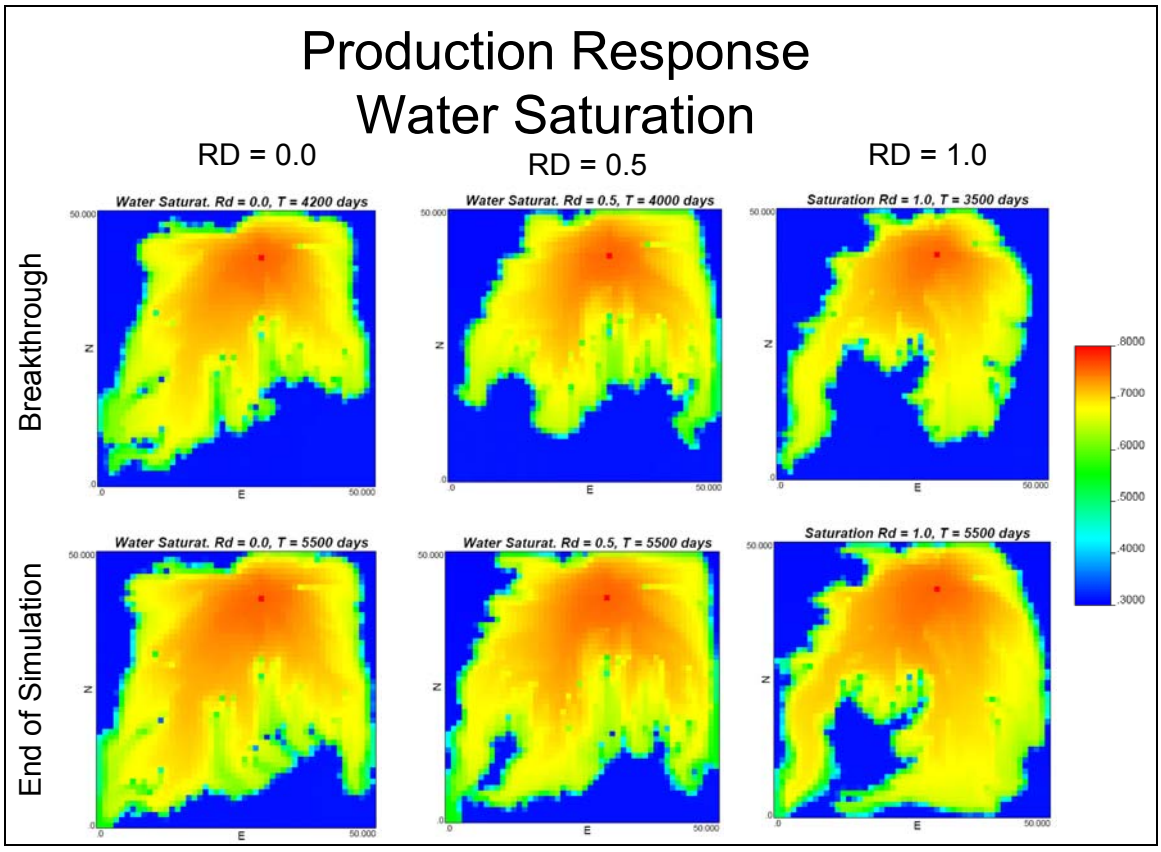


Figure 5: Sensitivity of water saturation distribution in the reservoir at 4000 days and 5200 days to r_D parameter.

The deformation parameter r_D is calibrated using the Dekker-Brent Iterative optimization procedure where the objective is to improve the fit between the flow response of the model (from the simulator) and the production history. The Dekker-Brent algorithm is an inverse parabolic interpolation method that has the great advantage of being a non-gradient based approach, and only requires the calculation of the objective function corresponding to different values of the deformation parameter. The calculation of gradients of the objective function with respect to the model parameters becomes prohibitively expensive, on repeated simulations. The Dekker-Brent Algorithm is used to estimate an optimal value of the deformation parameter, r_D^* (abscissa), corresponding to a minimum value of the objective function, $\Delta O = f(r_D^*)$ (ordinate). The algorithm require three abscissas, a, b and c with the corresponding values of the objective function f(a), f(b) and f(c); and b chosen such that a < b < c and f(a) > f(b) < f(c). The estimated

location of the abscissa (r_D^*) with the apparent minimum ordinate (objective function) is calculated by fitting a parabola through these three points.

$$r_D^* = \frac{\frac{(b+c)\,f(a)}{(a-b)(a-c)} + \frac{(a+c)\,f(b)}{(b-a)(b-c)} + \frac{(a+b)\,f(c)}{(c-a)(c-b)}}{2\left(\frac{f(a)}{(a-b)(a-c)} + \frac{f(b)}{(b-a)(b-c)} + \frac{f(c)}{(c-a)(c-b)}\right)}$$

The next figure illustrates the process of the inverse parabolic interpolation.

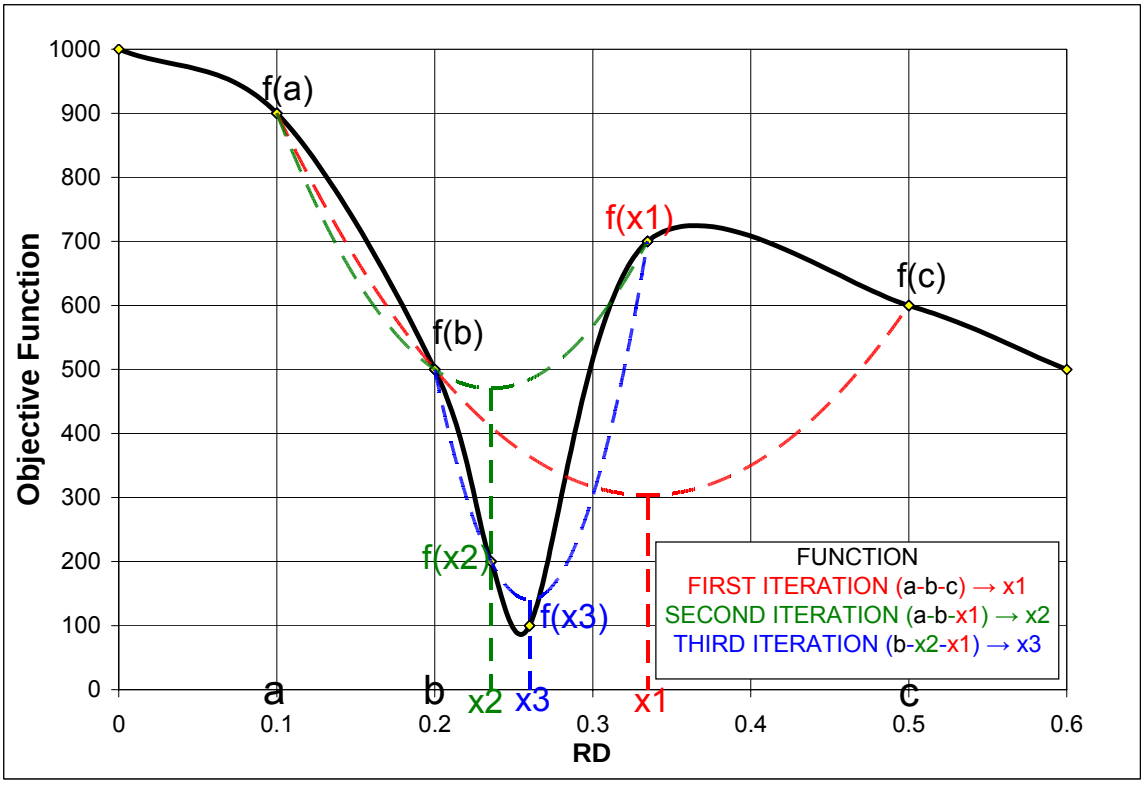


Figure 6: Variation in objective function value and the process of convergence to a minimum by the Dekker-Brent algorithm.

In this case, the objective function to be minimized is a measurement of the deviation between the simulated production response and the production history. Different production variables can be included in the objective function, including field and well pressures, single phase rates, two phase ratios, and basically any other variable that can be reported during the simulation, as long as there is a historical record to be compared

with. The individual variables are normalized to level their influence in the objective function. However, in some cases it might be useful to assign higher weights to some production variables in order to emphasize the relevance of their reproduction in the target model. The proposed objective function for N production variables on T time steps is:

$$Obj\ Fun = \sum_{t=1}^{T}\sum_{i=1}^{N}\left(\frac{Sim_{i,t} - Hist_{i,t}}{Hist_{i,t}}\right)^2$$

Where $Sim_{i,t}$ Represents the Simulated value of the production variable i at time t, and $Hist_{i,t}$ represent the correspondent Historical value for the same variable at the same time. The difference between the Simulated and the Historical values is normalized by the Historical value in order to control the influence of each variable in the objective function. However, this normalization method requires the exclusion of those historical values equal or close to zero, representing a drawback for the comparison of particular variables like the water cut. Other Objective functions can be easily implemented applying different norms and normalization methods.

The evaluation of the objective function requires the implementation of an interface between the geological modeling algorithm and the flow simulator. The geological modeling algorithm is the main program which includes all the tasks in the probabilistic approach for dynamic data integration, except for the flow simulation. It also includes the task of combining the conditional probability distribution P(A|B) derived on the basis of geology and conditioning data (indicator kriging) and the conditional distribution obtained by probability perturbation P(A|C) into a joint-conditional distribution

P(A|B,C). The geological model used in the flow simulator to calibrate the deformation parameter r_D is sampled from this jointly conditioned distribution, P(A|B,C). The methodology for obtaining P(A|B,C) from the elemental distributions P(A|B) and P(A|C) will be explained later in this document. The complete probability updating procedure therefore consists of:

- Performing indicator kriging and obtaining the conditional distributions P(A|B) at each unsampled location in the reservoir. Sample a realization from the P(A|B) by sequential simulation
- Corresponding to that realization of the permeability field and making an initial guess for r_D and obtaining corresponding P(A|C)
- Merge P(A|C) with P(A|B) to obtain P(A|B,C) and sample a realization from P(A|B,C)
- Perform flow simulation and obtain objective function. Revise estimate of r_D using Dekker-Brent approach and repeat until objective function is minimized

An important advantage of the probability perturbation method is that the complex non-linear inverse problem involved in the integration of dynamic information is simplified to a single (or few) parameter(s) optimization problem as compared to other methodologies that deal with the optimization problem using a very large parameter set.

The implemented interface between the geological modeling algorithm and the flow simulator is summarized in the following steps:

1. The geological modeling algorithm generates a file with a realization of the target model in the appropriate format. This file will be used by the flow simulator as an include file.
2. The modeling algorithm invokes the flow simulator.
3. The Flow simulator (ECLIPSE®) is run, generating an output file with the flow response of the realization.
4. The geological modeling algorithm reads the simulated flow response from the output file and evaluates the objective function.
5. The objective function is used into the optimization framework to generate a new realization of the geological model applying the gradual deformation approach.

Conditional Distributions Conditioned to Dynamic and Static Information

At this point, the methodology to estimate the local distribution conditioned to dynamic information, P(A|C), has been introduced, i.e. perturbing the local conditional distributions, with a parameter calibrated with the production history data. Yet, the contribution of the geological model of heterogeneity inferred from static information, P(A|B), needs to be honored in every realization of the reservoir model in order to ensure consistency with the geological model during all stages of the history matching procedure. Hence, probability distributions conditioned to dynamic information need to be combined with those conditioned to static information to obtain the global conditional distributions, P(A|B,C), from which the realizations of the target model are sampled.

The Permanence of Ratio Hypothesis (Journel, 2002) is the methodology applied to combine the individual distributions conditioned to each, dynamic and static information. The methodology is written in terms of ratios of probabilities, where the ratio is the relative distance to the occurrence of an event or a conditional event. The following distance or information measures a, b, c and x are defined:

$$a = \frac{1 - P(A)}{P(A)} \quad b = \frac{1 - P(A \mid B)}{P(A \mid B)} \quad c = \frac{1 - P(A \mid C)}{P(A \mid C)} \quad x = \frac{1 - P(A \mid B, C)}{P(A \mid B, C)}$$

The quantity a, for example, denotes the relative distance to the event A given P(A). If P(A) is one, the relative distance a is zero. The relative distance a is infinity, if P(A) is zero. The measures b, c and x can be interpreted similarly.

According to the permanence of ratios hypothesis, the relative updating of a geological event due to a dynamic event (C) remains the same irrespective of the presence of the static event (B). This can be written in terms of a,b,c and x as:

$$\frac{x}{b} = \frac{c}{a}$$

Consequently, the joint probability of the geological event given the dynamic and static information can be calculated from the ratios of probabilities of the geological event, the geological event given the dynamic information and the geological event given the static information.

$$P(A \mid B, C) = \frac{a}{a + bc}$$

In the approach, the marginal probabilities of the static, P(B), and dynamic events, P(C), which are difficult to estimate in practice, are not required.

Summarizing, the local distribution conditioned to static information, Prob(A|B), and the updated conditional distribution conditioned to dynamic information, Prob(A|C) are combined with the prior distribution, Prob(A) to obtain the distribution conditioned to static and dynamic information, Prob(A|B,C), utilizing the Permanence of Ratio hypothesis.

Development of a Robust Scheme for Parameterization of History Matching Schemes

The objective of history matching is to modify a prior model for the reservoir such that the updated model reflects the available production data and the uncertainties in production forecasts are reduced. The resultant geological model must therefore not only reproduce production data by numerical simulation but it must also be consistent with the geological knowledge of the reservoir.

The history matching process mainly consists of:

1) Identifying model parameters that could be modified to effect history match.
2) Defining a suitable objective function for the optimization procedure.
3) Proper selection and design of an optimization technique for reducing the objective function to a minimum.
4) Tracking computational cost associated with the flow simulations used within the selected optimization technique.

Many sets of parameter estimates may yield nearly identical matches of the data. A decision therefore has to be taken during the history matching procedure to determine the set of parameters estimates that are most appropriate given prior knowledge about the geology.

The data obtained from the field can be classified as static data or dynamic data. The static data do not vary with time e.g. permeability, porosity etc. while the dynamic data do vary with time e.g. production rates, well bore flowing pressures etc. The reservoir

model is developed based on the available static data using geostatistical simulation or a geological model. History matching techniques attempt to constrain the reservoir models to the dynamic data. The goal of the history matching is to minimize the **objective function** that measures difference between simulator response and the observed field data. The objective function may be defined as follows

$$\mathbf{Q} = (½)\sum_i \sum_j \sum_k \frac{(z_{ijk}^{observed} - z_{ijk}^{simulated})^2}{\sigma_{ijk}^2}$$

Z^{obs} is the measured value of a flow response such as bottom-hole pressure, gas oil ratio, oil production rates etc. $Z^{simulated}$ is the corresponding value obtained by performing a flow simulation on the reservoir model synthesized using the available data. σ_{ijk} is the measurement error. The index **k** means the type of observation data, **j** is the index for the number of wells, **i** is the index of the measurements dates for each well. The quantity $1/\sigma^2_{ijk}$ can be viewed as the weight assigned to the response Z_{ijk} – the larger the measurement error, the less the contribution of the mismatch to the overall objective function. *In the cases studied in this report, the measurement error is assumed to be the same for all the observations used in the calculation of the objective function.* Despite this simplification, the behavior of the objective function is strongly non-linear and small perturbation to model parameters can result in large fluctuations of the objective function.

Some authors have advocated that a prior information term (or regularization) should be added to the objective function formulation so that geological consistency is maintained during history matching process. Recently it has been proposed to add a 4D seismic data mismatch term in the objective function (Gosselin et al. 2000).

The combined objective function can then be written as:

$$\mathbf{Q} = \mathbf{a^*}\,(½)\sum_i \sum_j \sum_k \frac{(z_{ijk}^{observed} - z_{ijk}^{simulated})^2}{\sigma_{ijk}^2} +$$

$$\mathbf{b^*}\,½\,(y_k - y_k^{mean})^t C_p^{-1}(y_k - y_k^{mean}) \quad + \quad ½\,(s(m)-e)^t W_s (s(m)-e)$$

Where

a, b are the user's defined weight constants, W_s is the weight matrix assigned to seismic data. y_k is the history matching parameter (e.g. Permeability at a given grid cell), y_k^{mean} is the mean of the parameter. C_p^{-1} is the inverse of the covariance matrix of permeability and s(m) is the seismic derived predicted values and e are the observed seismic values. The first two terms in the objective function measure the production mismatch and the deviation from the prior geological model while the last term quantifies the deviation from the seismic information. During this course of study only the production mismatch term was used in the objective function.

Sensitivity Analysis

For the purpose of defining the domains the obvious choice would be to identify the sensitive regions to the objective function. Jacquard et al. (1965) first investigated the use of *sensitivity coefficients*. Sensitivity coefficients are defined as derivatives of the target simulation output with respect to parameters being adjusted to get a history match. The sensitivities may be calculated for one of the following:

1) Global objective function
2) Well specific objective function
3) Field phase (oil, water or gas) production rates
4) Well specific phase production rates.

Mathematically sensitivity coefficient may be defined as $\frac{\partial f}{\partial x}$ where, f is one of the simulator output listed above and x is the history matching parameter at a grid block, which is permeability for the cases presented in this report. In case the sensitivity were calculated with respect to multiple parameters x_i (e.g. porosity, permeability, thickness etc.), then normalization of the parameters would have to be performed in order to render the resultant sensitivities comparable on the same scale.

Sensitivity analysis done by perturbing parameter values and computing resultant changes in objective function values is highly inefficient as compared to methods using adjoint equations (Chen et al., 1971; Chavent, et al., 1975). For determining the sensitivity coefficients for one parameter, the adjoint equation method would take an additional 20% simulation time. Numerical perturbation would need one full simulation run in order to evaluate the sensitivity.

The program module ***Simopt*** in the ECLIPSE® suite has been used for the sensitivity analysis done in the report. ***Simopt*** uses the adjoint equation method for calculating sensitivity coefficients. All the flow sensitivity coefficients mentioned in this report are with respect to permeabilities in X direction. The results can be easily generalized to permeabilities in other directions.

Parallel Computing for History Matching

In parallel computing, a task is subdivided among many computer processors such that if done efficiently the total computational time scales with the number of CPUs assigned. Parallel computer architecture have generally been based on the following major concepts:
a) SIMD – Single instruction multiple paths.
b) MIMD – Multiple instruction multiple data paths

The MIMD architecture has emerged recently and is based on the principle that each CPU can operate independently, with frequent synchronization among the CPUs. MIMD architecture has three basic memory configurations:

– Flat shared memory: Allows access to all memory by each of the CPUs. This is costly and cannot scale well above 10 CPUs.
– Multi-level shared memory: Based on the concept of cache i.e. sophisticated software or hardware keep track of the current location of data within the global memory of the system.

– Distributed memory. Has become the most popular for massively parallel computers. These machines are based on the use of RISC-based CPUs, some with attached vector processors. Performance of each CPUs varies from a few million floating-point operations per second to more than 100 Mflops.

Recently the parallelization of reservoir simulators has been accomplished on distributed memory parallel computers. This parallelization has been accomplished on both MIMD and SIMD. A critical analysis of parallel computing with respect to reservoir simulation has been done by Killough (SPE 26634, 1993).

Message passing between the processors is critical to the performance of the distributed memory parallel computing. Two aspects of communication are important: **latency** and **bandwidth.** Latency may be defined as the time taken for setting up pathways between the processors and identifying locations that are involved in the transfer of data. In order to reduce the latency, several messages may be packed together, so that communication between the processors is reduced. In that case the bandwidth becomes critical.
Theoretically, for an N*N grid solved on P processors:

1) Total work time is proportional to N*N/P (Assuming complete parallelizing of the simulator)
2) Communication time is proportional to N

As the problem size increases:

1) Latency is affixed cost
2) Communication time increases with N
3) Work time increases with N*N

For inter-node connections, there are in general four options

a) Fast Ethernet
b) Gigabit
c) Myrinet
d) Quadrics

Band width and Latency for various switches in PC-Clusters configuration is as follows

Switch	**Bandwidth (Mbytes/sec)**	**Latency (microsecond)**
Fast Ethernet	12	150
Gigabit	128	26-12
Myrinet	421	7
Quadrics	400	5

The Myrinet and Quadrics have high bandwidth and are most suited for reservoir simulation. The research cluster used for this project has a gigabit switches between the different processing nodes. A reduction in parallel performance is therefore likely.

Amdahl's Law

The efficiency of a parallel program can be assessed by comparing the performance speed of a parallel algorithm with that of a program run on a single node. That measure known as speedup can be written as:

$$Speedup = \frac{1.0}{\left(s + \frac{p}{n}\right)}$$

s = serial fraction of program work including the communication overhead

p = parallel fraction of program work

n = number of parallel processors

To achieve better efficiency the subdivided tasks among the processors should be large as compared to the inter-processor communication. Amdahl's law implies that even the smallest portion of the model must be parallelized to achieve reasonable parallel efficiencies for a massively parallel processor with one hundred or more processors. In the notation of the above equation, making the fraction to be higher is better than keeping *s* to be high since the quantity *p* is divided by *n* the number of processors and the net gain in speedup may be considerable. Serial code sections become significant if a high number of CPUs (i.e. 100 + CPUs) are used in a single run.

It must be noted that parallel algorithms *are generally less optimized* than their serial counterparts and hence the performance of a parallel code running on a serial machine may often not be as good as the serial code running on the same serial machine.

There are certain challenges using parallel computation for reservoir simulation:

a) Firstly the non-recursive nature of existing linear solutions techniques for solving the sparse matrices encountered in reservoir simulation renders them unacceptable for massively parallel architectures. Considerable research is currently being focused on developing robust parallel linear equation solutions. Killough (SPE 26634, 1993) proposed a complex preconditioning scheme for conjugate residual type iterative methods such as ORTHOMIN.
b) The trade-off between load balancing and global data structure has yet to be thoroughly investigated.
c) Well and facility constraints and production optimization are generally implemented using serial algorithms and these may lead to severe serial bottlenecks.

One of the key issues affecting the performance of parallel computing is **load balancing**. Load imbalances can severely reduce the efficiency of parallel computing. If one of the processor within a parallel job is multi-tasking or has not finished the allotted load, then other processors within the parallel job will end up waiting. Hence parallel jobs **must** have dedicated resource (i.e. the processors to be used for parallel processing should not be mult-tasking or processing other jobs). The point being that if the speed of one processor slows down, other processors would be waiting reducing the efficiency of the tasks that are spawned on the other processors as well). Two common load-balancing methods are

– Static, data structures are allocated before beginning the computation
– Dynamic, restructuring of data structures as the computation proceeds

Wheeler and Smith (SPE 19804, 189) dealt with load imbalances brought about by irregularly shaped grids through a redistribution of active cells among the processors.

This technique works well in cases where the solution depends linearly on the number of grids blocks. But for cases where the solution is related non-linearly to the sub dimensions associated with the processor arbitrary redistribution of active cells can further exacerbate the problem. Killough (SPE 26634, 1993) distributed the computation work at each grid block within the simulation domain to all the processor nodes and assigned to each node its proportion of the calculations. A substantial improvement appears to be possible but unresolved issues still exist:

a) The number of grid blocks will seldom be a perfect multiple of the number of nodes.
b) The number of iteration will vary for each grid block. This implies that to achieve better load balancing, a prior idea of number of iterations and hence allocation of work to the computer nodes is necessary.
c) The communication overhead may overshadow the performance of the algorithm.

Killough (SPE 29102, 1995) later proposed a receiver-initiated dynamic load-sharing algorithm to achieve high parallel efficiencies. It adapts the workload on each node in a dynamic way and can react to any sudden perturbation that can appear during the simulation. The algorithm suffers from the fact that there is significant loss in performance for a small number of processors since one processor is dedicated for supervising and monitoring the computation at the remaining nodes.

Distributed Computing

In a distributed configuration, the processors do not share memory or clocks. Distributed computing systems group individual processors together via network connections and pool the computing resources in order to accomplish CPU intensive computation. The network provides a means by which client and host machines communicate, sharing computed information or passing information that is required by the computations. Local area networks (LAN) of PCs connected by Ethernet connections are ubiquitous and are good examples of distributed computing. Distributed computing requires coordinating the efforts of a collection of processing nodes linked together by a network. The common tool used for the external parallelization is PVM (Parallel Virtual Machine). PVM allows

the development of the programs that can send “slave” tasks to different *cpus* in a network.

A strategy to apply distributed system to solve large-scale reservoir simulation problems has been demonstrated by Yu et al. (2002).

The use of distributed computing/external parallelization to speed up history matching procedures has been studied by Schiozer and Sousa (SPE 39062, 1997). They showed that sensitivity analysis could be performed efficiently, as well as various direct search optimization techniques can be improved, by the use of external parallelization (or distributed computing). The choice of optimization techniques to be used is quiet important. Methods that use derivatives may have convergence problems. For this reason, Leitao and Schiozer (SPE 53977, 1999) recommends direct search methods that are most robust. Quenes et al (SPE 29107, 1995) applied parallelization techniques to history matching reporting good results. The benefits of external parallelization could be summarized as follows:

– To calculate sensitivities
– To determine the best search directions for direct optimization techniques.
– To launch several simulations in direct search methods.

There are some advantages of using external parallelization (or distributed computing) over parallel computing

– Efficiency of current commercial simulator codes is maintained.
– An existing network of workstations can be used without the need of high investment in parallel computers or in communication.

The main disadvantage is slow data transfer imposed by the LAN communication protocols, but in this kind of work, actual computations are so time consuming that communication time can be neglected for practical applications.

Evaluation of existing parallel simulators

Two commercial simulators Eclipse® and VIP® were tested for their parallel computation capabilities. Both have the parallel as well as the flux boundary options available. The flux boundary option is important from the standpoint of distributed computing. The choice of simulator was dictated by the availability of options (outputs) such as the Hessian matrix of the objective function that would help in identifying the sensitive regions. *Simopt* - a history-matching module in Eclipse, has the option to report sensitivities. The commercial simulator VIP doesn't have similar options. Keeping in view this limitation of VIP, the following section dwells on important features of the Eclipse flow simulator that facilitate parallel computation and retrieval of flux boundary conditions.

Eclipse (General)

The Eclipse simulator suite consists of two separate simulators: Eclipse 100 Specializing in black oil modeling, and Eclipse 300 specializing in compositional modeling. Eclipse 100 is a fully-implicit, three phase, three dimensional, general purpose black oil simulator with gas condensate options. Eclipse 100 can be used to simulate 1, 2 or 3 phase systems. Two phase options (oil/water, oil/gas, gas/water) are solved as two component systems saving both computer storage and computer time. Eclipse 300 is a compositional simulator with cubic equation of state, pressure dependent K-value and black oil fluid treatments. Eclipse 300 can be run in fully implicit, IMPES and adaptive implicit (AIM) modes.

Fully implicit technology (**Black Oil**)

Eclipse (Black oil) uses the fully-implicit finite difference method to ensure numerical stability over long time steps. The non-linear fully-implicit equations are solved precisely by reducing all residuals to user set tolerances. Newton's method is used to solve the non-linear equations. The Jacobian matrix is fully expanded in all variables to ensure quadratic (fast) convergence. Most simulators cannot apply fully implicit methods to large problems. In Eclipse, these restrictions are removed by nested factorization, which solves large problems efficiently and reliably.

Adaptive Implicit and IMPES (**Compositional**)

In a compositional model where the number of components and hence equations to be solved is greater than say 5 or 6, the cost of performing fully implicit simulations can become prohibitive in terms of both memory and CPU time. In Eclipse 300 this problem is tackled by using an adaptive implicit (AIM) scheme, making cells implicit only where necessary. In Eclipse 300 adaptive implicit, fully implicit or IMPES solution techniques may be selected. For larger compositional studies AIM can be selected.

Parallel Option

The Parallel option allows a single simulation job to be distributed across a number of processors. This allows large simulations to be carried out in shorter time than would normally be the case with the standard simulators. The results obtained using a number of processors will generally agree with single processor solutions within limits of engineering accuracy. The scalability is poorer than one might expect, as the linear solver becomes less efficient when larger numbers of processors are utilized. Eclipse 100 first partitions the reservoir either in the x or the y direction, depending on the outer solver direction. The code is optimized so as to automatically divide the reservoir into domains with approximately equal numbers of active cells.

For reservoirs with a significant number of inactive cells, the default partitioning is not sufficient to load-balance the domains. It is possible to control further the way in which cells are assigned to domains by applying the following formula:

$$\text{Active cell load balance \%} = \frac{\text{total active cells in reservoir}}{max(\text{active cells per partition}) \times \text{number of procs}} \times 100$$

The one important difference between Parallel Eclipse 100 and Parallel Eclipse 300 is in the approach used for the parallelization of the linear solver. Eclipse 100 uses a modified nested factorization approach in which a 1-dimensional domain decomposition is performed, full coupling of the solution across the reservoir for each linear iteration is maintained, at the expense of increasing the work required per linear iteration by 1.3.

Eclipse 300 allows a 2-dimensional domain decomposition (For large numbers of domains, a two-dimensional decomposition is desirable, **to avoid thin strip domains)** approach.

Flux Boundary Conditions

Flux boundary conditions enable to runs to be performed on a small section of a field using the boundary conditions established from a full field run. These smaller field simulations can be distributed to the different nodes of a cluster. Flows across the boundary of the reduced field are written to a FLUX file at each min-time step of the full field run. The flux file is input to the smaller field simulation as the boundary condition for that simulation.

Instead of using the flows of each phase from the full field run as boundary conditions on the reduced run, an alternative treatment (i.e. alternative set of boundary conditions) using pressure and saturations is available in Eclipse. When **more severe changes** are made to the flow regime and fluid distribution, the pressure boundary conditions should be used to allow the phases to re-distribute adjacent to the boundary and allow the switch from inflow/outflow in the full field run to outflow/inflow in the reduced run (i.e. the reverse flow directions to account for abrupt changes in reservoir fluid distribution).

Theory of Principal Components Analysis

The original motivation for defining reservoir zones on the basis of sensitivity coefficients came from typical history matching applications. As mentioned previously, during history matching, the permeability of each grid block is adjusted in order to obtain a simulated response close to the observed response. This implies that the optimization problem can be quite large (potentially equal to the number of grid blocks) for a field scale simulation model. It is just not efficient to search a parameter space that is so large and this has been the bane of traditional methods for history matching.

An approach for solving this problem is to group correlated grid blocks together. Different correlation measures can be defined and these include both static data based measures (e.g. the variogram of permeability, geobodies etc.) as well as dynamic measures (based on flow criteria). The flow-based measures account for flow across geologically or statically defined geobodies due to the imposed boundary conditions. Besides, since flow response characteristics constitute the objective function in the history matching procedure, it is sensible to base the identification of flow regions based on the dynamic correlation characteristics of the medium. When defining the regions, it is also important to realize that full independence between the different domains within a reservoir is not possible unless there are regions separated by zero permeability streaks or zero transmissibility faults. Nevertheless, an attempt can be made to isolate domains on the basis of least correlation in terms of their effect on the objective function.

One scheme of identifying the regions on the basis of the high sensitivity and least correlation was presented by Robert Bissel (4^{TH} European conference on Mathematics of Oil Recovery, Norway 1994). In that method, pilot points are identified and the sensitivity coefficients at the pilot point locations stored in the Hessian matrix is also calculated. The Hessian matrix is defined as follows:

$$S = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial y}\right) & \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial z}\right) & \cdots \\ \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial x}\right) & \frac{\partial^2 f}{\partial y^2} & \frac{\partial}{\partial y}\left(\frac{\partial f}{\partial z}\right) & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

where f is the objective function and x, y, z are the parameters (permeabilities at the pilot points). The Hessian matrix is a measure of the sensitivity of the objective function to the history matching parameters (permeabilities at different grid blocks). It also measures the

interaction of the parameters in different parts of the field and the resultant effect on the objective function.

From a history matching perspective, the properties of a good Hessian are:

1) Large diagonal elements This means that objective function is sensitive to the parameter at a particular location and is insensitive to the parameter value at neighboring locations. If small changes in the parameter values causes large changes in objective function, the value of the parameter corresponding to minima can be found accurately
2) Similarly sized diagonal elements –This is required so that the matrix is well conditioned and small changes in the data do not cause large changes in the solution.

The use of the Hessian matrix directly within optimization based history match approaches has been discussed in the literature. For this project, since the goal is to implement a fully probabilistic approach that relies on probability update parameters r_D defined locally for different regions of the reservoir, the Hessian matrix is analyzed with an objective to determine reservoir regions. Several techniques for analyzing the Hessian matrix in order to define reservoir regions have been discussed in the literature and these are reviewed next.

Factor Analysis

Factor analysis is a multivariate data reduction method. The factors are constructed in a way that reduces the overall complexity of the problem and takes advantage of inherent interdependencies in the data. As a result a small number of factors will usually account for almost the same amount of information as do a much larger parameter set. Factor analysis creates a minimum number of new **hypothetical variables** such that the new variables contain most information.

Mathematically stating, the parameter set $x_i, i = 1,.., p$ is expressed as a linear combination of factors f_j, $j = 1,.., k$:

$$x_i = a_{i1} f_1 + a_{i2} f_2 + ... + a_{ik} f_k + e_i$$

$$x_{i+1} = a_{i+1,1} f_1 + a_{i+1,2} f_2 + ... + a_{i+1,k} f_k + e_{i+1}$$

$$\vdots$$

$$x_p = a_{p,1} f_1 + a_{p,2} f_2 + ... + a_{p,k} f_k + e_p$$

In the notation of linear algebra, the above mathematical model can be succinctly expressed as:

$$X_{(N*P)} = F_{(N*K)} A'_{(K*P)} + E_{(N*P)} \qquad (1)$$

Where X is the original parameter matrix, F is the matrix of factors, A the matrix of the factor loadings (or weights) and E is a matrix of residuals. K is the scalar used to denote the reduced set of factors to be used. K is always less than P, the number of original parameters. The number K is chosen such that the resultant factor set captures most of the original variance in the original parameter set. Higher the redundancy in the original data matrix the lesser the size K required for representing the set amount of variance.

In the notation of Equation (1), it is assumed that we have N observations of each parameter X_i, i = 1,..,p. Each column of the matrix F has N elements representing the varying contribution of that factor towards the N realizations of the parameter. Hence, each factor is some unobservable attribute of the objects and F is the totality of such attributes for the objects. In reservoir modeling terms, the matrix X would have deviant scores of a permeability realization as a row vector. Deviant score of permeability at a particular location in the reservoir may be defined as the permeability obtained by subtracting the average permeability of the realization from the permeability value corresponding to that grid location.

The factor loading matrix, A, contains the coefficients that must be used to combine the factors to obtain a particular value of permeability at a particular location. A column vector of A may be regarded as containing coefficients that describe the composition of the factors making up the original parameter, permeability in this case. A factor is thus a linear combination of the observed variables, a concept that is used in principal components analysis.

Eigenvalues and Eigenvectors

There is a classical mathematical proof which states that for a symmetric Grammian matrix (which has nonnegative eigenvalues) $S_{(p*p)}$ of rank r, a symmetric Gramian matrix $T_{(p*p)}$ of a given lower rank k < r can be written that approximates S in a least squares sense. Mathematically stating

$$S = \lambda_1 u_1 u_1^{'} + \lambda_2 u_2 u_2^{'} + ... + \lambda_r u_r u_r^{'} \quad (2)$$

$$T = \lambda_1 u_1 u_1^{'} + \lambda_2 u_2 u_2^{'} + ... + \lambda_k u_k u_k^{'} \quad (3)$$

$\lambda_1 > \lambda_2 > ... > \lambda_k > ... > \lambda_r$ are the eigenvalues and $u_1 > u_2 > ... > u_k > ... > u_r$ are the respective eigenvectors.

This Least-Square property of eigenvalues and eigenvectors has been used for the factor analysis of a covariance matrix. The covariance matrix is a positive definite matrix and hence Grammian matrix. Comparing the mathematical expressions (2) and (3) with the expression(1), it can be seen that the loading function a_i are akin to the eigenvalues λ_i and the factors can be constructed on the basis of the eigenvectors u_i.

Geometrical interpretation of eigenvalues and eigenvectors

In statistical jargon, a bivariate scatter diagram depicts the joint variability of a pair of variables. In the case of bivariate Gaussian distribution, equal density contours drawn on the bivariate plot are in the shape of ellipses. If the variables are uncorrelated, the ellipse is replaced with a circle and on the other hand if the variables are perfectly correlated, a line results. For more than two variables, the data points can be viewed as forming p-dimensional hyperellipsoid. Pearson (1901) and later Hotelling (1933) realized that the major and minor axes of the hyperellipsoid are identified by the eigenvectors of the covariance matrix. Since the principal axes of the hyperellipsoid point in the direction of maximum variance, the placement of eigenvectors in multi-domensional space may be viewed as a problem of rotation of axes according to the criteria of maximum variance. The eigenvectors of the covariance matrix are uncorrelated in the sense that they are linearly independent and are sorted in the order of decreasing variance contribution. An eigenvalue of zero indicates that the corresponding minor axis of the hyperellipsoid is of

zero length. This suggests that the dimensionality of the space containing the data points is less than the original space.

There are numerous methods mentioned in the factor analysis literature for determining the number of eigenvectors to be extracted and retained for representing the variability of the original phenomena. Some of more practical methods are described below:

1) The proportion of the variance extracted by the eigenvector. The proportion can be obtained by dividing the corresponding eigenvalue with the trace of the Hessian matrix. Any threshold may then be applied on the amount of total variance (or the total information) extracted, say 80%. The real utility of PC analysis lies in the fact that few factors (eigenvectors) would extract most of the information. Thus, if there are 180 eigenvectors (corresponding to the Hessian matrix obtained from 180 pilot points), then the first few (not more than 5 in most cases) would serve the purpose of representing most of the variability exhibited by the Hessian.
2) The size of the factor loading may be used as the basis.
3) The significance of the residuals: Residual matrix may be calculated after extracting each eigenvector. The point at which no further factors need to be extracted is reached when the residual matrix consists of correlations solely due to random error.

Rotation of factor axes

Describing the data in terms of eigenvectors is akin to rotating the hyperellipsoid plotted using a original set of orthogonal axes and aligning it to a new set of orthogonal axis defined by the eigenvectors. The orthogonal eigenvectors point in the direction of maximum variance. However, it is important to realize that there can be other criteria for the rotation of axes. In fact there are an infinite number of sets of vectors that will describe the configuration of data as well. One of the objectives of factor analyses could be to display the configuration of the variables in as simple a manner as possible without worrying about maximum variances or orthogonality etc.

In case there is a lot of overlap of the components of eigen components among the eigenvectors i.e. the eigen component of two or more eigenvectors are high at a particular grid cell, then separation into sub-domains on the basis of maximum variance may not be possible and we may have to resort to alternate system of rotation of factor axes. Mathematically, it is simple matter to rotate one set of the axes to another set, provided that the angle of rotation is given. The major task is to determine the angle through which the axes must be rotated.

Kaiser (1958) developed varimax method of rotation, which is one of the most popular methods of rotation of axes. A "simple" factor would be one having a few high loadings (eigen components) and many near zero loadings. This can be determined by specifying an angle of rotation such that the variance among the eigen components is maximized. To avoid sign complications, the variance of the squared eigen components is maximized. For the entire matrix of factor loadings, simplicity is attained when the sum of each individual factor variance attains a maximum.

$$S_T = (1/p) * (\sum_{j=1}^{k} \sum_{i=1}^{p} b_{ij}^4 - (1/p) * \sum_{j=1}^{k} (\sum_{i=1}^{p} b_{ij}^2)^2) \quad (4)$$

Where, b_{ij} is an element in the rotated matrix, p is the number of eigen components and k is the number of eigen vectors. Kaiser suggested that each row of the matrix should be normalized to unit length before this variance is computed. Working with normalized rotated elements b_{ij}, the following expression for variance has to be maximized:

$$S_{JL} = p\left(\sum_{i=1}^{p} \left(\frac{b_{ij}}{h_i} \right)^4 + \sum_{i=1}^{p} \left(\frac{b_{il}}{h_i} \right)^4 \right) - \left(\sum_{i=1}^{p} \left(\frac{b_{ij}^2}{h_i^2} \right)^2 + \sum_{i=1}^{p} \left(\frac{b_{il}^2}{h_i^2} \right)^2 \right) \quad (5)$$

Where, h_i is a normalizing factor. The rotated elements b_{ij} are obtained by applying the following transformation:

$$\begin{aligned} b_{ij} &= a_{ij} \cos\theta_{jl} + a_{il} \cos\theta_{jl} \\ b_{il} &= a_{ij} \cos\theta_{jl} + a_{il} \cos\theta_{jl} \end{aligned} \quad (6)$$

a_{ij} are the original loadings. The optimum value of θ_{jl} that result in maximum variance is obtained by implementing an optimization procedure:

- Substitute expression (6) in expression (5) and differentiate with respect to θ_{jl} and set derivative to zero.
- Determine θ_{jl} for each possible pairs of j and l. The matrix of rotated loadings then can be obtained from:

$$B = AT_{12}T_{13}...T_{jl}... \quad (7)$$

Where T_{jl} represents the transformation matrix derived from the rotation of the factors j and l.

The procedure described above is applicable for establishing variable groups when multiple realizations of parameter sets are available. In the context of this project, the factor analysis methodology outlined above is used to determine groups of locations that exhibit similar sensitivity characteristics. Some case examples demonstrating the applicability of the proposed methodology are discussed in the Results section.

EXECUTIVE SUMMARY

The goal of history matching is to obtain a model of a reservoir from which reliable forecasts of future production can be obtained. This project approaches the problem of developing a robust scheme for history matching with two guiding ideas. First, the reservoir model that results from the scheme should be geologically plausible. That is, the nature of permeability heterogeneity in the model, including correlation lengths and structures at multiple length scales, should be consistent with what is known about the geology of the reservoir. The motivation for this condition is that the inverse problem represented by history matching – choosing a large set of parameters (local values of permeability) so that a small set of data (flow rates at wells as a function of time) is matched – is under-constrained. A solution to this inverse problem that makes geological sense is more likely to provide reliable forecasts. The second guiding idea is that it must be possible to obtain insight from a computer implementation of the history matching in a practical length of time (e.g. overnight). The time scale for decision making in many industrial applications does not allow for lengthy calculations.

We are implementing these ideas in this project by taking advantage of two technologies: a probabilistic approach for integrating production data into the reservoir model, and parallel/distributed computing for speeding up the turnaround time for forward simulations.

Our probabilistic approach to dynamic data integration is based on conditional probability distributions that account for the uncertainty in permeability at any given location in the reservoir. At the beginning of the history matching process, these distributions embody the statistical properties of the reservoir heterogeneity as inferred from well logs, core samples, etc. The key idea being tested in this project is that the conditional probability distributions are iteratively updated during the history matching procedure to account for the additional information contained in the dynamic response data (the production histories). We have implemented this update with a perturbation parameter r_D, that controls the magnitude of the deformation applied to the values of permeability in the model. Four schemes for this perturbation have been developed,

yielding an approach that ensures a gradual deformation of the model. This has proved to be a good way to maintain consistency between the model and the geological reality. To date we have demonstrated this scheme with a commercial simulator (Eclipse®) to perform the forward flow simulations for each permeability realization.

We have extended the notion of a single perturbation parameter r_D to consider a set of such parameters for a given reservoir. Each parameter applies to a particular domain in the reservoir; the domains are non-overlapping. This extension provides a basis for decomposing the flow simulation onto parallel/distributed computing platforms as well as for increasing the effectiveness of the perturbation scheme. For the latter, we have used a sensitivity analysis. The influence of the value of permeability at any given location upon the production data predicted from the forward model can be extracted from the simulator. We have applied a principal components analysis of these sensitivities to identify sub-regions within the reservoir domain that have the most influence on the flow behavior, and therefore are the most important regions to consider when varying the permeability values to effect a history match. Trials of this method on realistic reservoir test cases yielded encouraging results. The sub-regions are also natural choices for decomposing the physical domain into sub-problems for distributed or parallel computing. We have tested these ideas with two commercial simulators (Eclipse and VIP) for which parallel versions are available.

We have implemented a rudimentary integration of all these ideas into a single history-matching algorithm. Only preliminary tests have been carried out so far, but the results suggest that the approach will yield the desired properties: a robust method that finds a geologically consistent reservoir model in a reasonable amount of wall-clock time. This is a good foundation for the next phase of the research.

EXPERIMENTAL

In this project we are developing algorithms and implementing them on computers. The scope of work does not include experiments.

RESULTS AND DISCUSSION

Development of a Robust Scheme for Parameterization of History Matching Schemes

Sensitivity Analysis

In order to demonstrate the sensitive analysis procedure, a deterministic reservoir model representative of a fluvial depositional environment has been used. The model has 120,000 (100*120*10) grid blocks with high permeability channels. The channel distribution and orientation exhibit small changes from one layer to the next. The process for generating the synthetic data set including the calculation of pseudo-seismic responses has bee discussed in Mao, 2000. This particular reservoir model was used for many of the cases discussed later in this report and is referred to as the ***STAN5*** data set throughout the report.

Example 1

Computing the sensitivity coefficient corresponding to the permeability in each grid block is practically not feasible due to the associated computational cost. The sensitivity analysis is done at 180 uniformly distributed pilot points. The sensitivity coefficients are then spatially interpolated to all other grid blocks. Since the different layers of the reservoir exhibit strong resemblance to one another, the strategy adopted for the analysis was to utilize one of the layers as the reference layer and the other layers as the incremental realizations for calculating the required sensitivities. For this case, the fifth and the third layers of the reservoir were used for computing sensitivities while the history was generated using the fourth layer. The history is generated for 500 days. Adjacent layers are chosen deliberately so that the initial guess is near to the actual

model. There are *three producers and two injectors*. Other details are included in the data file attached in Appendix A1.

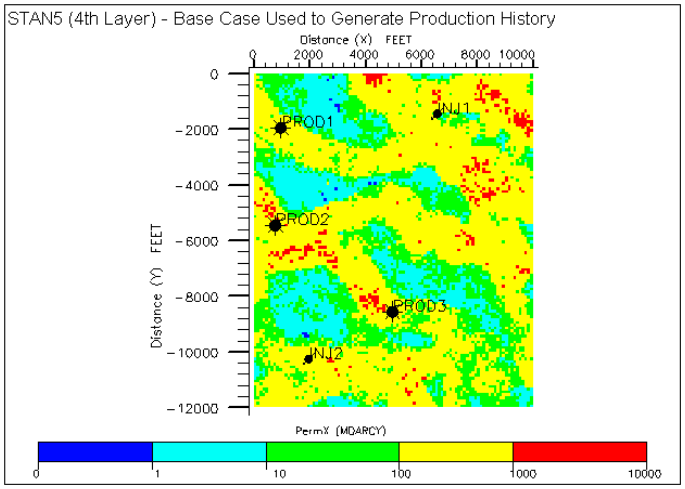


Figure 7: Reservoir model utilized to generate the base case production history

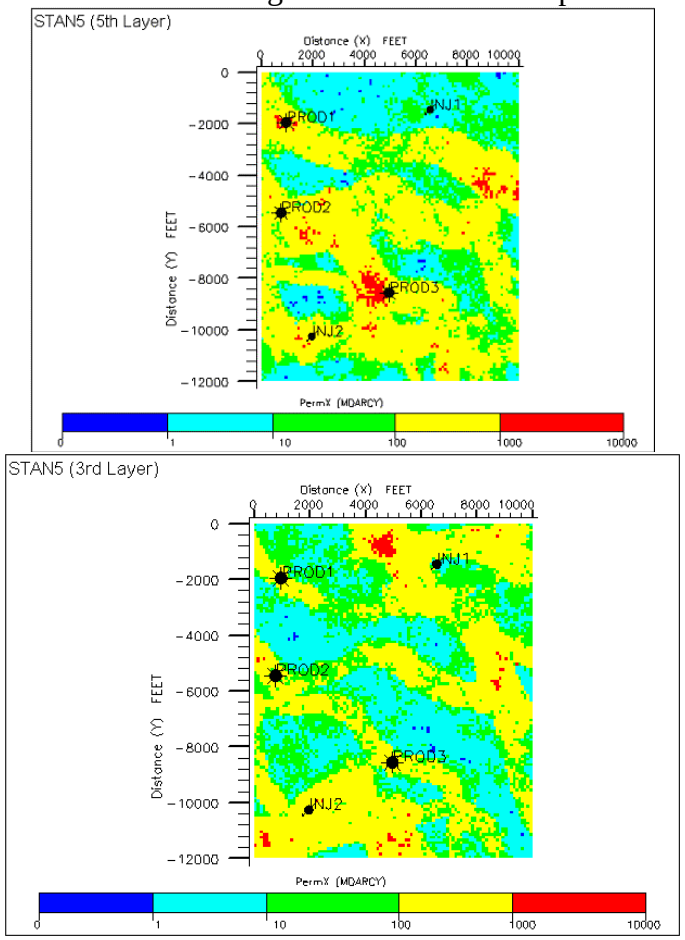
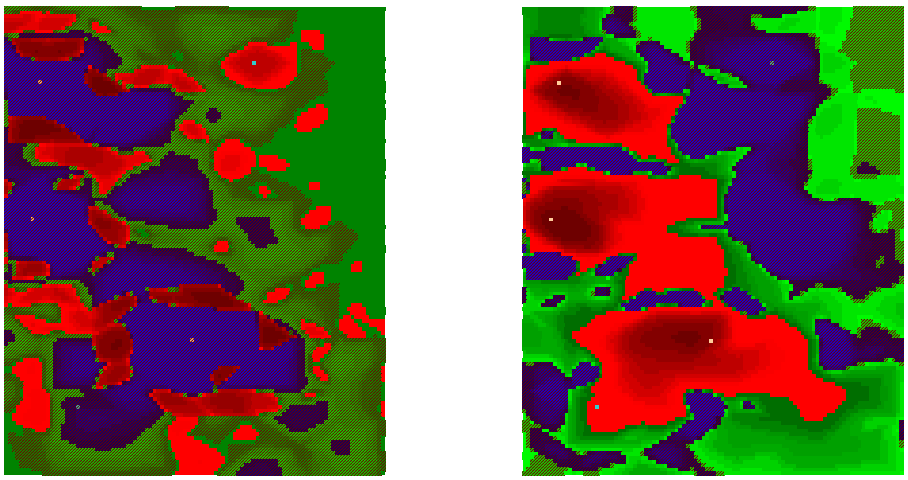


Figure 8: The pair of permeability models used for calculating the sensitivity coefficients.

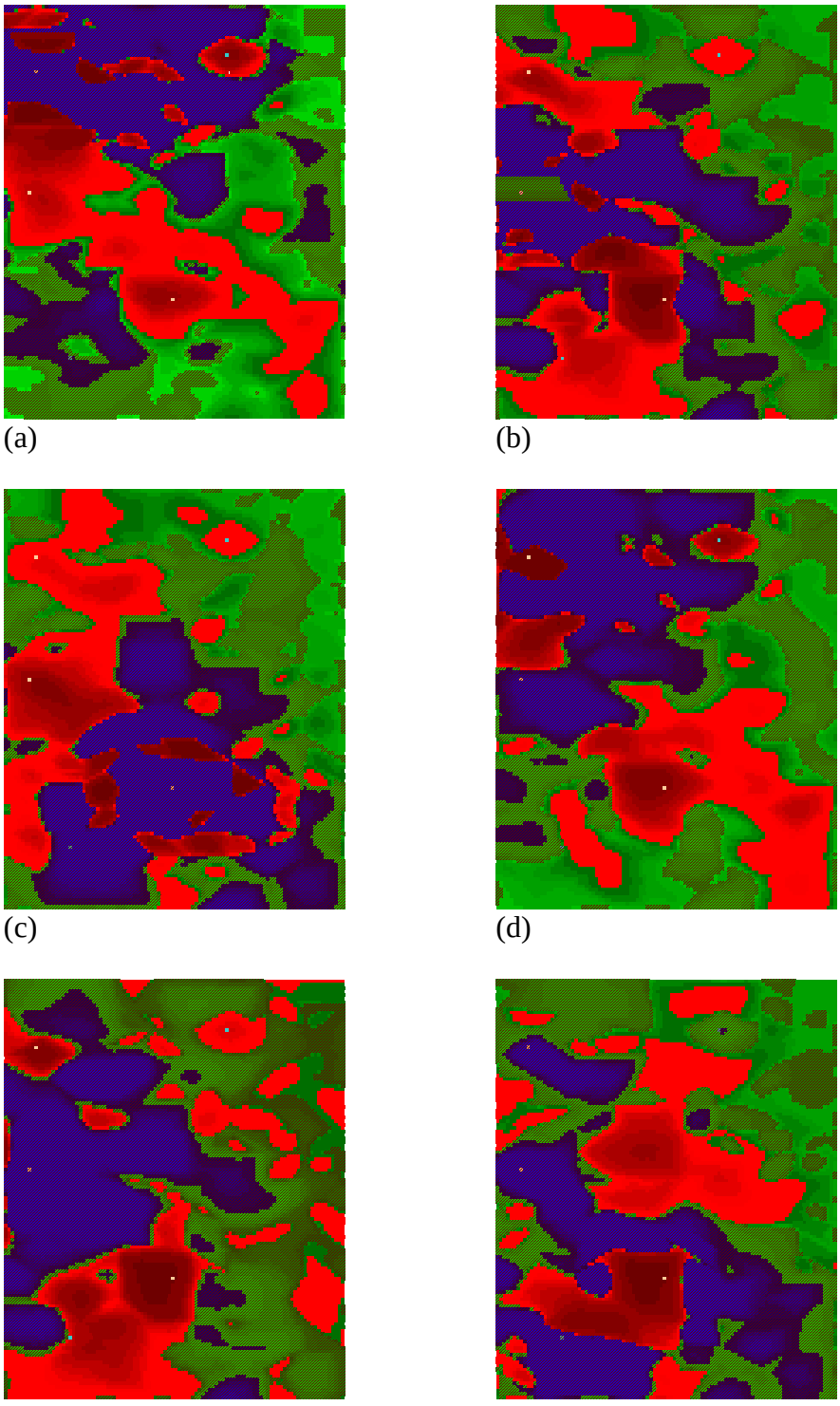
The following Figure 9 shows the global (derivative of global objective function with respect to permeabilities) sensitivity coefficients. The degree of change is proportional to absolute magnitude while the direction of change is governed by the sign of sensitivity coefficients. The highest values are represented in red, the middle ones in green and the lowest in blue. The negative values are shown with the shaded lines. Negative sensitivity implies that permeability values in those regions need to be increased in order to match the flow response for the base case.



(a) (b)

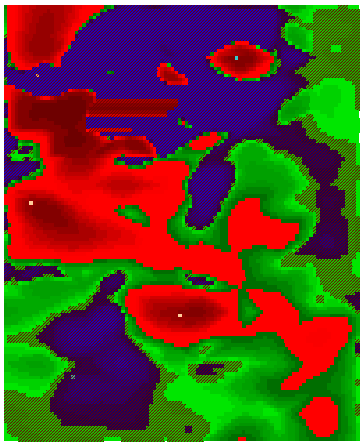
Figure 9: Global sensitivities for: a) 3rd layer, and b) 5th layer. The permeability corresponding to the 4th layer is used as the base model for calculating the sensitivities.

Sensitivity coefficient can also be calculated on the basis of individual well response. The following Figures 10 and 11 depict the sensitivity of different well responses to the permeability variations in layer 3, given the 4th (STAN5) layer as the base case.

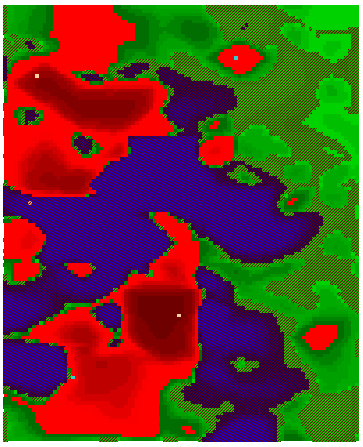


(a) (b) (c) (d) (e) (f)

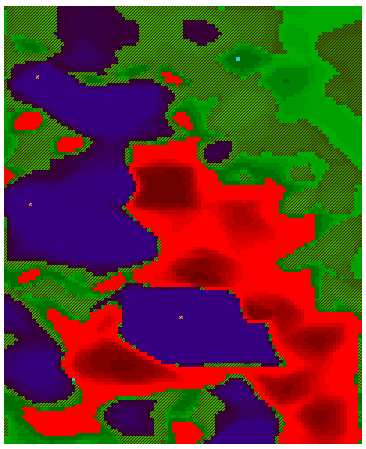
Figure 10: Sensitivity coefficients for the layer 3 permeability model calculated on the basis of: a) Well 1 oil production rate; b) Well 2 oil production rate; c) Well 3 oil production rate; d) Well 1 water production rate; e) Well 2 water production rate; and f) Well 3 water production rate.



(a)



(b)



(c)

Figure 11: Sensitivity coefficients for the layer 3 permeability model calculated on the basis of: a) Well 1 gas oil ratio, b) Well 2 gas oil ratio, c) Well 3 gas oil ratio.

The following observations are based on the results in Figures 10 and11:

- Decreasing objective function defined on a per well basis may not be a wise approach for history matching. There are regions in the reservoir, that show positive sensitivities for one well and negative sensitivity for another well. This implies that decreasing the permeability in one region may decrease the objective function for one well but it may increase the objective function for another well.
- Even for a particular well it is possible to decrease the mismatch for particular phase production rate at the cost of the increasing the mismatch for another phase. This is illustrated by the sensitivity regions defined on the basis of oil production and water production.
- The sensitivity regions are consistent with the geological structure of the reservoir. The results indicate that local changes in permeability within the channel facies would result in maximum decrease in objective function value. However, it is also to be noted that such crisp delineation of the underlying geology may be a direct

consequence of the close resemblance of the layer 3 structure to the base case layer 4 model.

Based on the results and observations made above, it is proposed that a better approach to defining reservoir regions may be using the global objective function. The global objective function would be a weighted sum of individual phase production rates and would hence represent a compromise between competing flow response characteristics.

Example 2

In this case the permeability realization is very different from the base case unlike the earlier case where the initial guess closely resembled the base case (reference) model. Since an iterative, permeability updating procedure for history matching is proposed in this research, the objective is to gauge the frequency with which the reservoir zones designation would have to be updated during the iterative procedure.

The base model is still the **4th layer of STAN5** and all other inputs remain the same. Sensitivity coefficients are calculated for two models. The first model has perm-y and perm-z equal to the base case while the perm-x is very different from the base case and is shown below. The second model is a constant permeability field with permeability being 1000 md in all directions.

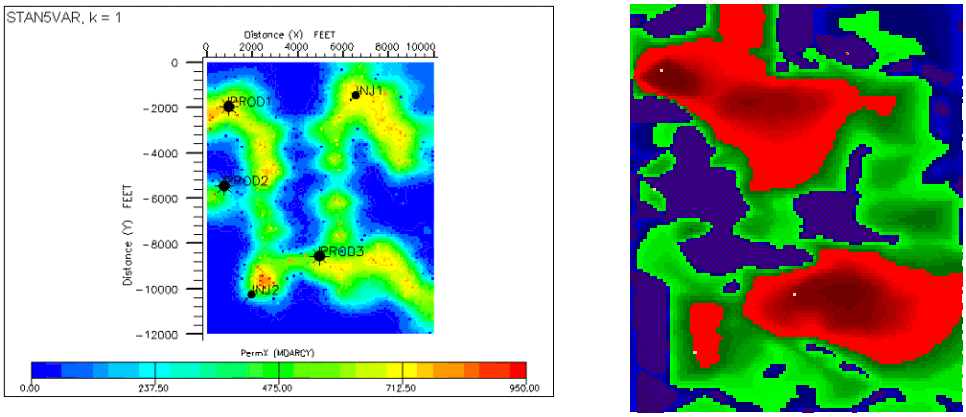


Figure 12: Permeability field with channel in the same place as in the base case model, but the channel permeability is reduced. The corresponding global sensitivity coefficients computed on the basis of the base case model are shown in the right.

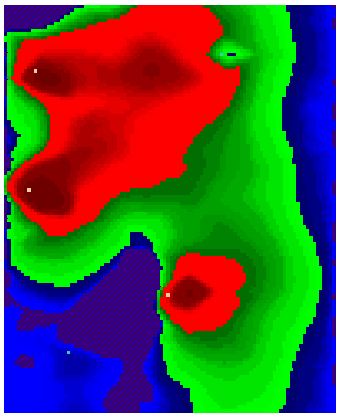


Figure 13: Global sensitivities corresponding to a homogeneous permeability field.

In the absence of prior geological knowledge (Figure 13), the results suggest that the most appropriate approach to effect a history match is by increasing the permeability in the near well regions. In Figure 12, where there is reservoir continuity indicated by the presence of channels, reduction in objective function is obtained by increasing the permeability in some regions (maybe away from wells) and reducing the permeability in some other regions. In both these cases, the results seem to indicate that the sensitivity coefficients are not so reliable if the initial guess is not near the actual model**.**

This important observation can be understood on the basis of the following conceptual plot.

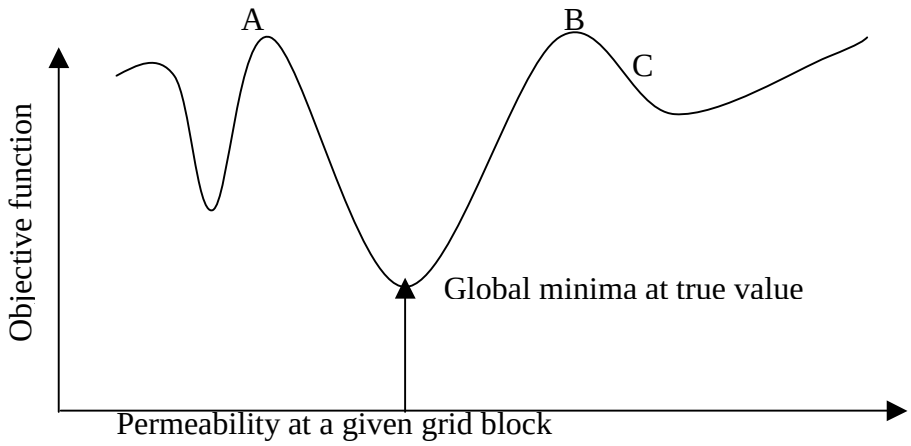


Figure 14: Typical objective function profile and the importance of a good initial guess.

The sensitivity analysis would give reliable information if the initial guess is between points A and B. The negative slope of the objective function from A to the global minima

would suggest a negative sensitivity i.e. the permeability value in the grid block would have to be increased. If the permeability value becomes too high, the positive slope of the objective function profile from the global minimum point to B would suggest a positive sensitivity coefficient value and hence a decrease in permeability value to result in a history match. However, if the initial guess had been C, the negative slope of the objective function profile at C would suggest a negative sensitivity value and hence a increase in permeability and hence a further drift away from the true minimum value.

Test Cases

The factor analysis procedures described earlier can be applied on the basis of the correlation exhibited by flow sensitivities, porosity and pay thickness and/or reservoir permeability. The domain decomposition obtained using each of these criteria are presented in this section.

Domain decomposition based on flow sensitivities

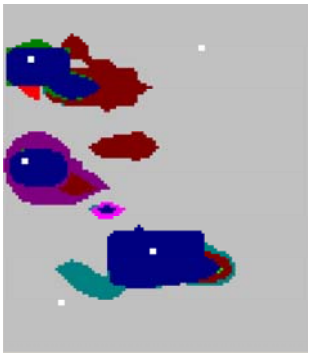
Following procedure is followed for obtaining the domains based on principal component analysis of the Hessian matrix:

1) Obtain eigenvalues and corresponding eigenvectors of the Hessian matrix. Eigenvectors thus evaluated would describe an n dimensional state, where n is the number of the pilot points (or the rank of the Hessian matrix).
2) Interpolate the component of the eigenvector throughout the reservoir from the component values at the pilot points.
3) Rank the eigenvalues based on their magnitude. The eigenvalues will always be positive since the Hessian matrix is positive definite.
4) Apply a threshold on the size of domain covered by first eigenvector. It could be say 40% of the total grid blocks. In other words all the grid blocks for which the *absolute value* of the eigen component corresponding to the first eigenvector exceeds the P60 value (of the distribution of eigen components) are grouped together as the first domain.
5) Follow step 4 for all the remaining eigenvectors with respective thresholds. During this step the grid block cells covered by the earlier (higher in rank) eigen components are excluded.

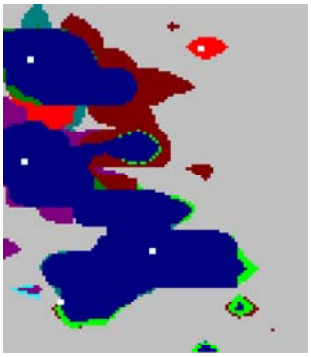
The number of eigenvectors retained would depend upon the amount of variance deemed necessary to be reproduced. An appropriate criterion for selecting the number of eigenvectors to be retained is still being tested. In principle, the thresholds should neither be too large which increases the correlation among the domains nor too small which leads to ineffective history matching since the identified domains will be too small.

The above procedure for delineating reservoir domains was implemented and the results presented share the following color scheme in all the ensuing figures:
First eigenvector positive components, negative components.
Second eigenvector positive components, negative components.
Third eigenvector positive components, negative components.
Fourth eigenvector positive components, negative components.
Fifth eigenvector positive components, negative components.

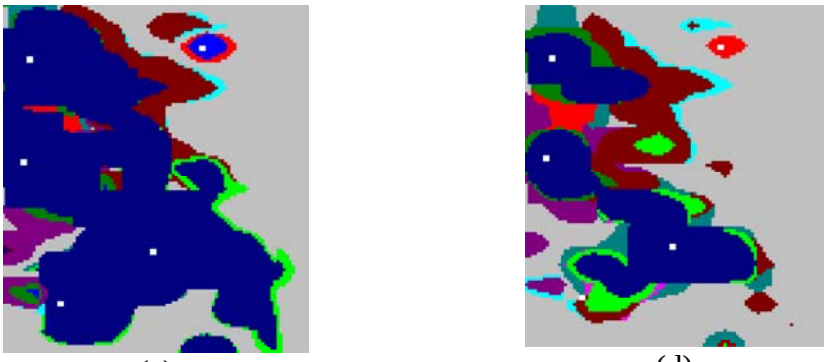
The example below is based on the Principal Component analysis of the Hessian matrix derived from the 3rd layer of STAN5. The production history is generated using the 4th layer.



(a)



(b)



(c) (d)

Figure 15: a) Reservoir regions delineated by applying a 10% cut-off volume i.e. each eigenvector covers 10% of total grid blocks. The regions covered by the second and subsequent eigenvectors may not cover that percentage of grid blocks due to the overlap with higher-ranking eigenvalue, b) Regions obtained by applying a 30% threshold, c) Regions obtained by applying a 50% threshold, and d) By applying different cut-off volumes to different eigenvectors: 20%, 25%, 30%, 35%, 40% cut-offs applied on the first, second, third, fourth and fifth eigenvectors respectively.

In all the examples shown above, the information content in the eigenvalue is just used to rank the vectors. The results seem sensitive to the cut-off volume specified for identifying the regions corresponding to each eigenvector. A possible solution to reduce this dependency on cut-of thresholds may be to use the eigenvalues to scale the magnitude of the eigenvectors. Recall that in Step 4 of the algorithm, all grid blocks for which the *absolute value* of the eigen component corresponding to the first eigenvector exceeds the P60 value (of the distribution of eigen components) are grouped together as the first domain. When this procedure is implemented, it is possible that a particular grid cell is assigned to a particular eigendomain because its eigencomponent exceeded the P60 value corresponding to that eigenvector. But the same grid cell may have an eigen component corresponding to a lower ranked eigenvector that is more than that for the higher ranked eigenvector, but that location has already been marked to belong to a particular eigen domain. This causes the size of the domain corresponding to the higher ranked eigen-vectors to swell and that corresponding to the lower ranked eigenvectors to be too small and thereby making it necessary to select different thresholds for different eigenvectors. Instead, if scaled eigenvectors are used then it is unlikely that significant overlaps between eigen components for different eignvectors would occur and so one common threshold may be used to obtain all eigen-domains.

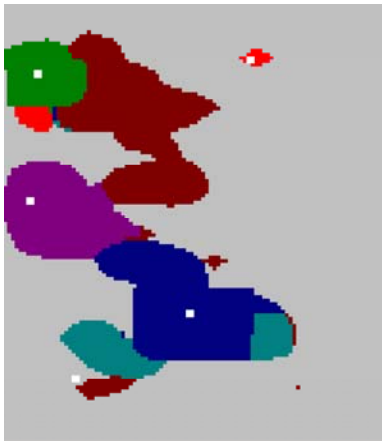


Figure 16. Domains delineated in the layer 3 of the STAN5 data set using scaled eigenvectors. The regions identified represent a 30% cut-off volume.

In the results shown in Figure 16 there seems to be less overlap of regions after scaling of eigenvectors.

Domain decomposition based on porosity and thickness

Domain delineation based on principal component analysis of the Hessian matrix is computationally expensive. It may be argued that in lieu of calculating reservoir domains on the basis of flow sensitivities that could be expensive and extremely dependent on the particular permeability realization used to perform the flow simulation, we could utilize the porosity and thickness model for the reservoir to delineate reservoir zones. In most situations, it could be argued that porosity and reservoir thickness can be fairly reliably ascertained using auxiliary data such as seismic.

In order to test this conjecture, an attempt was made to delineate the domains based just on the porosity and thickness at the grid blocks. Geobodies were defined on the basis of the product of porosity and thickness. Geobodies represent connected volumes of the reservoir that have storativity ($\phi \cdot h$) values greater than a threshold. The definition of geobodies was later modified in order to take into account the connectivity of storativity to a well location. In this revised scheme, all regions within a search radius around wells where the product of porosity and thickness exceeds a threshold are grouped in a domain. The search radius is made proportional to the upper flow rate constrains on the wells.

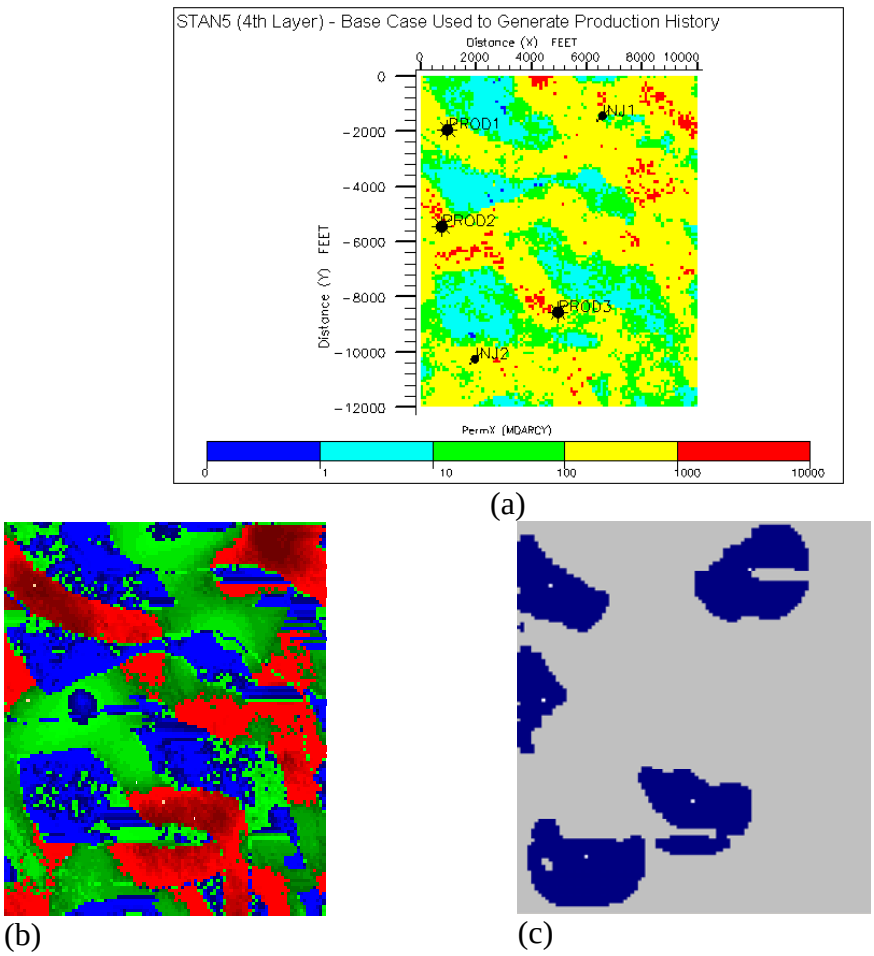


Figure 17: a) Base case model (Slice 4 of STAN5) used to generate the production history; b) Geobodies computed using the porosity and layer thickness information for Slice 4; c) Geobodies computed using a modified definition that focuses on a search radius of 16 units around well locations.

Comparing the results in Figure 16 and Figure 17(c) it can be seen that the shape of the reservoir domains in the vicinity of wells appears fairly consistent in both the models. However, the interwell connectivity regions observed in Figure 16 are missing in Figure 17(c). Updating permeability values in these interwell regions is likely to significantly influence the history match.

It could be argued that the reservoir regions delineated in Figure 16 would be strongly dependent on the characteristics of the permeability realization and since the permeability model is likely to be very uncertain prior to history match, the robustness of the delineated regions is likely to be questionable. In order to test this argument, the

sensitivity based domain delineation procedure was applied for the case where a homogeneous permeability is assumed. That map shown in Figure 18 shows that though the shape of the reservoir regions is likely to change significantly during the process of history matching, the interwell connectivity regions continue to be important sensitivity regions and the configuration of wells and flow rates help define these connectivity regions.

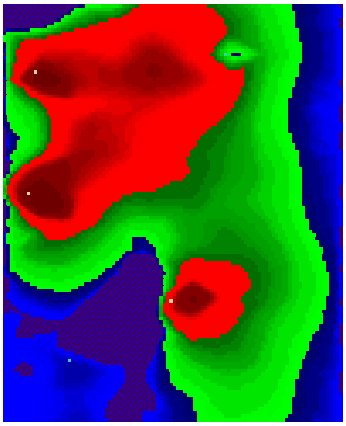


Figure 18: Sensitivities computed for a homogeneous (1000 md permeabilities in X, Y and Z directions) permeability field.

Domain decomposition based on permeability covariance matrix

Principal component analysis can also be applied to the covariance matrix describing the permeability variations in the reservoir. Once the eigenvectors are obtained appropriate thresholds in the manner described above may be applied to get the domains. It is expected that the domain delineation procedure using the covariance matrix would identify regions that exhibit similar permeability structure (geological facies). A potential drawback could be that the analysis is just based on the static data. It is possible that for particular boundary (well) conditions, there might be significant flow across facies boundaries, resulting in connected flow-based sensitivity regions but un-connected permeability-based regions. Conversely, permeability covariance in regions significantly removed from the well regions could still be identified as important reservoir regions based on PCA of the permeability covariance matrix, although their influence on the reduction of the history match objective function may be minimal.

A synthetic 100*120 permeability field was used to demonstrate the procedure for delineating reservoir regions using the covariance structure of the permeability field. The reservoir was assumed to consist of three facies layers with equal proportion (4000 grid blocks each) of locations belonging to a particular facies. This is deliberately done to evaluate the efficacy of eigenvectors in identifying the facies. An exponential covariance structure was assumed for each facies, with the prior variance (sill) in facies 1 equal to 800 md^2, facies 2 equal to 600 md^2 and facies 3 equal to 1000 md^2. The correlation length is assumed to be 170 units within each facies. In the vertical direction (across facies), the correlation length is reduced to 20 units and the corresponding prior variance is assumed to be 100 md^2. A cut off 33% was applied to first three eigenvectors. The covariance matrix for this case is of the following form:

The eigen domains identified by the principal component analysis procedure are shown in Figure 19 below. As can be seen, the eigenvectors could resolve the three facies perfectly. The facie with highest variance is being identified as most sensitive, the facies with the second highest variance as the next most sensitive and so on.

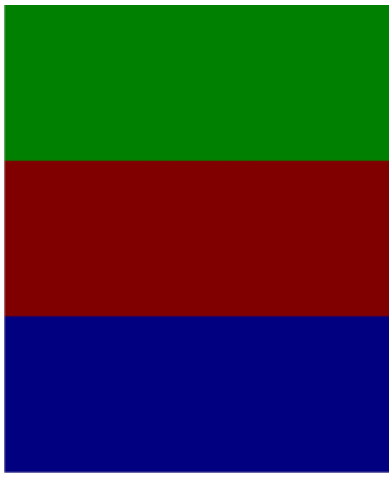


Figure 19: Figure showing the eigen domains based on covariance matrix of permeabilities. The following is the color scheme employed: First eigenvector positive components , second eigenvector positive components and third eigenvector positive components .

A similar exercise was carried using 2nd, 3rd, 4th, 5th and 6th layer of STAN5. The covariance matrix was calculated from the above-mentioned 5 layers. The covariance between any pair of nodes in the reservoir domain was obtained using pairs of values at

those nodes over the 5 layers. The layers of the reservoir are therefore taken to be multiple realizations of a common stationary process and hence the covariance is calculated by employing a stationarity decision over the multiple layers of the reservoir. It must be noted that a covariance value calculated over a sample set of 5 is likely non-robust, nevertheless the calculation was performed to demonstrate the methodology.

.

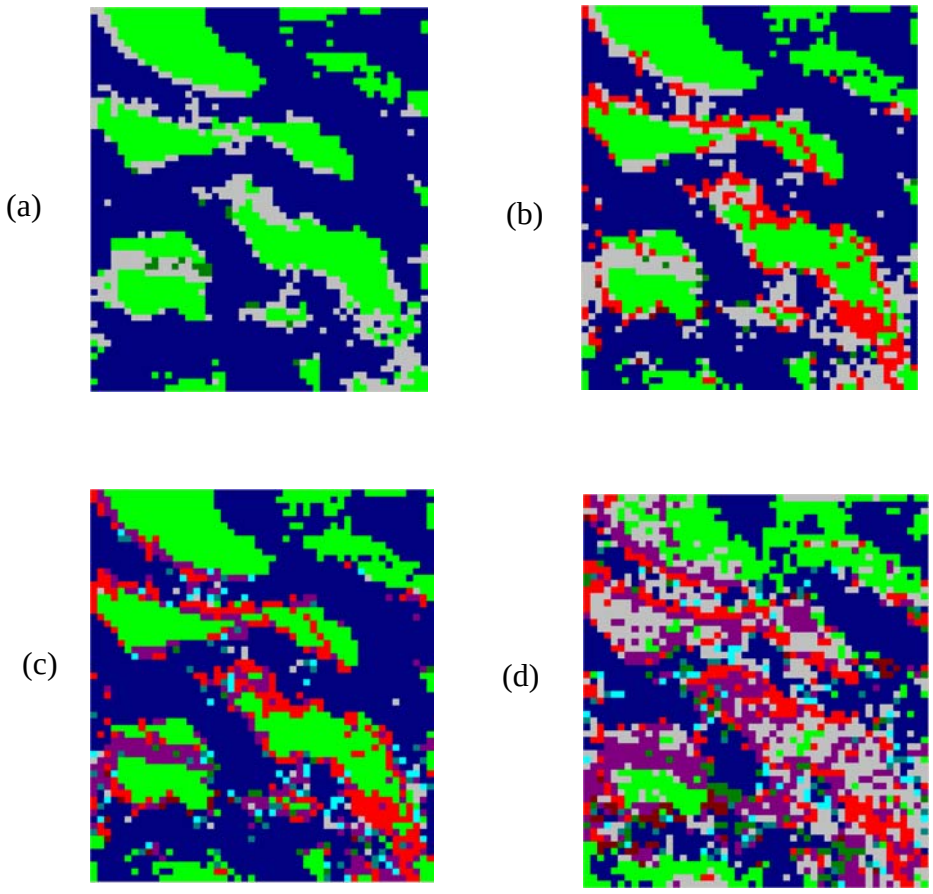


Figure 20: Reservoir regions for the STAN5 data obtained using the permeability covariance matrix: a) Applying a 60% cut-off for the first two eigenvectors of, b) Applying a 50% cut off on the first 3 eigenvectors, c) Applying a 50% cut-off on the first 5 eigenvectors, d) Applying a 30% cut off on the first 5 eigenvectors

As seen in Figure 20, the delineated zones retain the imprint of the channel feature common to all the reservoir layers. Increase in the number of eigenvectors retained to represent the variability and/or reduction in the cut off threshold, adds more noise to the resultant maps.

In order to evaluate whether the lack of adequate statistical mass influences the calculation of the covariance matrix and hence the determination of eigen domains, the same exercise was performed on 50 realizations of a permeability field generated by the p-field simulation technique. The reservoir size is 50 x 50 grid blocks. The results shown in Figure 21 were obtained.

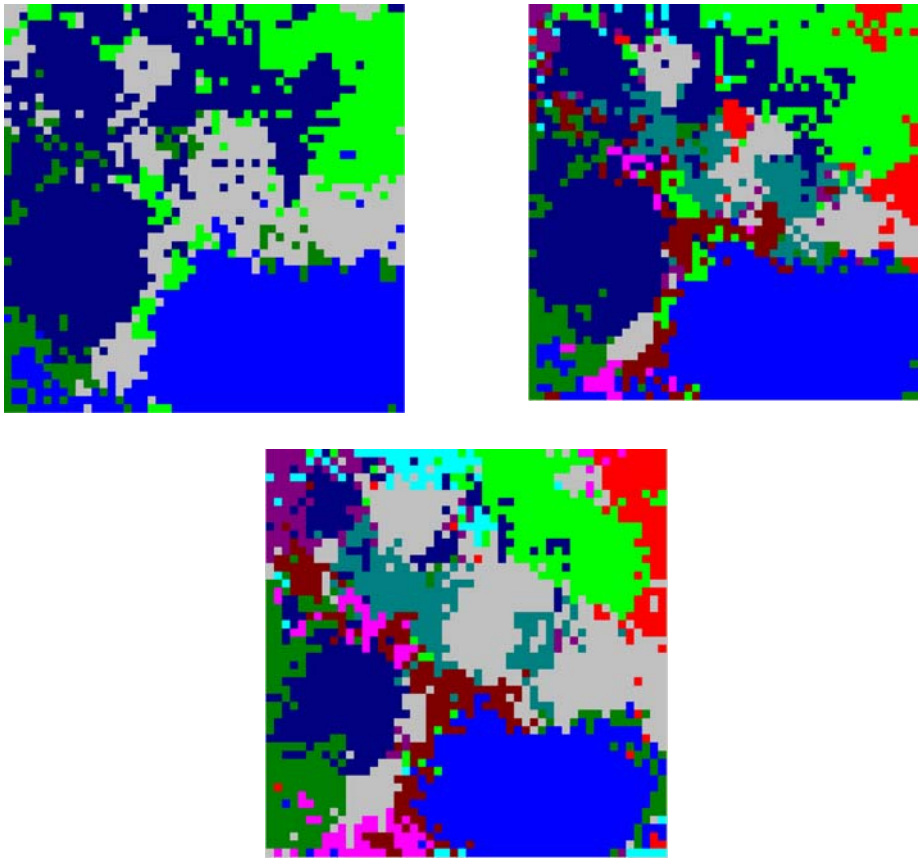


Figure 21: Reservoir regions delineated corresponding to a suite of permeability models obtained using p-field simulation. The eigen decomposition of the permeability covariance was performed and the reservoir regions were obtained by: a) Applying a 60% cut-off for the first two eigenvectors, b) A 50% cut off applied to the first 5 eigenvectors, and c) A 30% cut off applied to the first 5 eigenvectors.

It can be seen in the figure above that the choice of the cut-off threshold can have an important influence on the shape and size of the delineated zones. If the threshold is large then a number of grid nodes get assigned to regions spanned by the higher ranked eigenvectors and as a result the regions corresponding to the lower ranked eigenvectors are small.

In the procedure described above, the reservoir regions are delineated based on the permeability variations observed over many permeability realizations while sensitivity coefficients from flow are calculated for a given realization. Another important observation is that the domains defined based on the principal component analysis of permeability covariance matrix are such that the most uncertain regions of the reservoir (with maximum variance) are calculated to have the highest eigen components and hence deemed most sensitive for history matching purposes. However, regions with the most uncertain parameters may not always be the most sensitive regions. For example, the permeabitities at particular reservoir regions may be most uncertain but if those locations are far away from wells, then they will contribute less to flow and hence may not be consequential to the objective function.

Computational cost associated with sensitivity analysis

As earlier mentioned calculation of sensitivity coefficient can cost as much as 20% of the total run time. Some approximations are necessary in order to keep the computational cost reasonable. The feasibility and robustness of reservoir regions defined by restricting the duration of the flow simulation runs was evaluated. The results in Figure 22 indicate that while the sensitivity coefficients do change in the magnitude with respect to time, their relative magnitudes remain the same provided the permeability, well constraints, number of wells remains the same. The sensitivity coefficients defined on percentile scale remain almost the same. This implies that reservoir regions defined by performing the flow simulation for a fraction of total simulation period would serve the purpose of defining reservoir regions.

It has to be emphasized that in this project, the objective of sensitivity analysis is only to identify reservoir regions. History matching would ultimately be performed using a probabilistic approach only. The absolute magnitudes of the sensitivity coefficients are not important since they are not used in the history matching process.

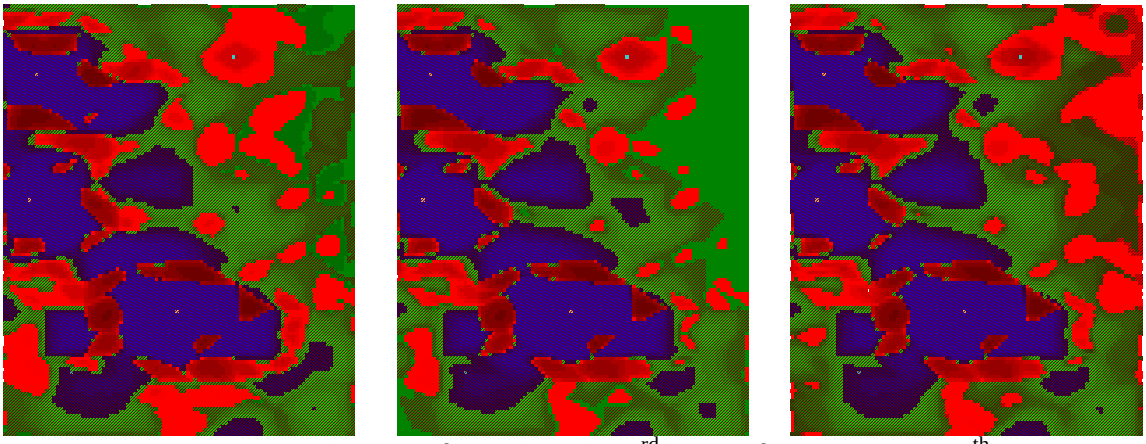


Figure 22: Sensitivity analysis performed on the 3rd layer of STAN5 (with 4th layer as base case for generating history) at a) 100 days, b) 500 days, and c) 1500 days.

Rotation of eigenvectors as per the Varimax criteria

The procedure for determining the rotation angle that yields the maximum variance of eigen components (Varimax) was tested. Figure 23 (a) shows the results of the sensitivity analysis performed on the 3rd layer of STAN5 with the base case production history data generated for the 4th layer. The first four eigenvectors are plotted each with 30% cut-offs. In that figure, there are some regions that are masked by higher ranking eigen vectors (e.g. the bright green regions that is masked by the blue colored eigen domain). Figure 23 (b) shows the first four eigenvectors rotated as per the varimax criteria each with 30% cut-offs. The variances are more evenly distributed in the case of regions obtained after variamx rotation. Net variance extracted remains the same for two cases but sensitivity constraint (i.e. the first eigenvector should align along the axis of maximum variance of hyper-ellipsoid) is compromised for the varimax case in order to get less overlap among the eigen components. The feasibility of applying eigen rotation to determine reservoir regions exhibiting least overlap is thus demonstrated.

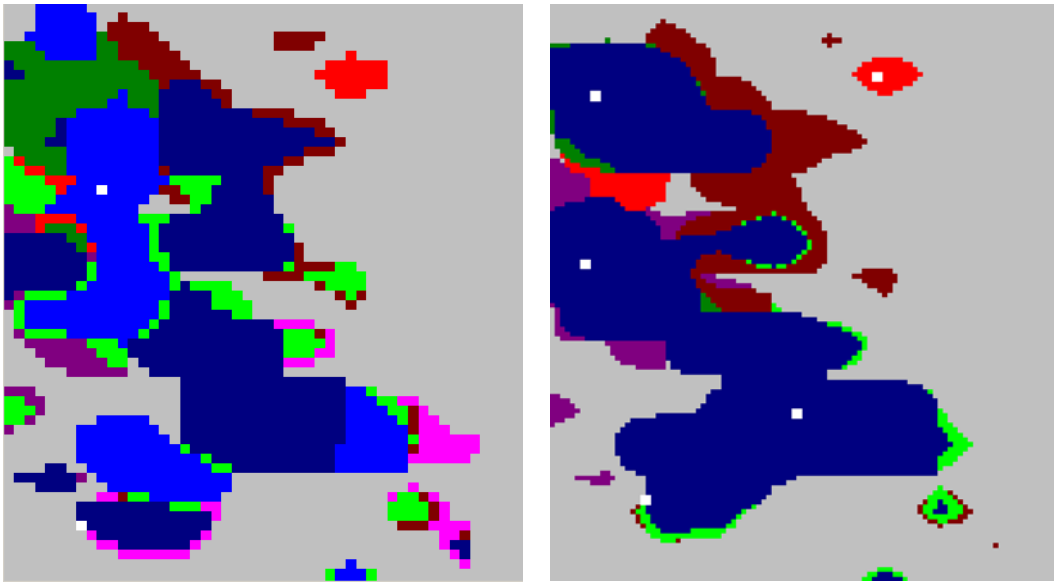


Figure 23: Reservoir regions delineated corresponding to the 3rd layer of STAN5: a) Using conventional PCA and eigenvectors aligned along the principal axes of the hyperellipsoid, b) After rotating the eigen components using a Varimax (maximizing the variance of eigen components) criteria.

A Scheme For History Matching Using Domains In The Distributed Computing Environment .

The property of the domains determined by applying the PCA procedure is that the regions exhibit least correlation in terms of their influence on the objective function. Thus, each domain can be perturbed on a different node of the cluster while aiming for the minimization of the global objective function. Finally the perturbed regions can be all put together and this would be justified since the perturbed domains are assured to be least correlated. The proposed approach is particularly suited for distributed computing since independent tasks of equal magnitude can be performed on multiple cpu nodes. This would amount to flow simulations of the same realization at different nodes while perturbing different regions. Rather than doing a full field flow simulation on the whole reservoir domain, certain low sensitive regions can be excluded based on the results of the sensitivity analysis. All the remaining gird location in the reservoir will be subject to

flow simulations on multiple nodes. At each node only the probability distribution corresponding to a particular domain will be perturbed – all nodes outside the domain retaining the original unperturbed probability distributions. Preliminary code development for implementing this scheme is already underway.

Parallel Computing for History Matching

The proposed approach for history matching utilizes reservoir regions determined by the sensitivity analysis procedure and subsequent perturbation of the local conditional distributions describing the uncertainty in permeability values in that region in order to effect a history match. The perturbation to the probability distributions are performed using an updating parameter termed r_D. There can be two plausible ways of handling multiple "rds" corresponding to multiple domains i.e. parallel or distributed computing. Each has its merits. For parallel computing the first requirement is that the domains be of comparable size. This problem can be addressed by having two different definitions of domains i.e. that updating is done on the domains defined as per the sensitivity criteria while during the running of simulator the domains divided are defined such that they are equal in size. This implies that the flow simulation can consist of multiple r_D regions, and so the problem of r_D optimization of multi-parameter optimization. The resultant algorithm will therefore be complicated, defeating the very objective of the proposed history matching approach.

Using distributed computing, the same problem of r_D optimization can be approached using the 1-D optimization methodology that renders the proposed method efficient. Each domain can be perturbed at different node of the cluster while targeting the global objective function. Finally the perturbed regions can be put together since the domains

perturbed were least correlated. The proposed approach is particularly suited for distributed computing since independent tasks of equal magnitude need to be performed. This would amount to flow simulations of the same realization at different nodes while perturbing different regions. Rather than doing the flow simulation on whole realizations at different nodes, certain low sensitive regions may be excluded by using boundary conditions. These low sensitivity boundary regions would not need frequent updating since the there would be minimal flow across those regions. In the distributed environment, any commercially available serial flow simulator can be utilized at each of the computational node and furthermore, the high efficiency of serial algorithms can be put to use.

Test Cases

Parallel Simulation

In order to demonstrate the parallel flow simulation capabilities of Parallel ECLIPSE 100, the following cases were run.

For Case1 and Case 2 a 4000 ft. x 4000 ft. x 20 ft reservoir model was assumed (with constant permeability of 500 md, constant porosity of 30%). The total number of grid blocks in the reservoir was assumed to be 48,000. The reservoir is assumed to have one vertical producer in the middle of the reservoir. Four parallel processors were utilized for the parallel simulation. Parallel Eclipse 100 was used that allows the reservoir to be divided in either X or Y dimension. Three cases were run, with identical reservoir dimensions and well locations. The simulation was run for 500 days. All parallel simulations utilized four processors. In our case the definition of aspect ratio is the number of grid blocks in x to y directions. The whole point of doing the following cases was to demonstrate that certain schemes of domain decomposition could be computationally efficient. In the later stages of the algorithm (after distributed computing part is over), the whole reservoir is subjected to one ‘rd’ optimization so as to remove any artifacts (if any). This information would then be useful to as to have computationally efficient domain decomposition scheme.

Case 1) Aspect ratio changed while the number of grid blocks are kept same

The effect of the aspect ratio of the reservoir on the performance of the parallel simulation was assessed.

Grids	Aspect Ratio	Time (Sec)	Producer Coordinates
4*4000*3	0.001	184	(2,2000)
16*1000*3	0.016	75	(8,500)
20*800*3	0.025	56	(10,400)
40*400*3	0.1	144	(20,200)
80*200*3	0.4	236	(40,100)
160*100*3	1.6	322	(80,50)
400*40*3	10	491	(200,20)

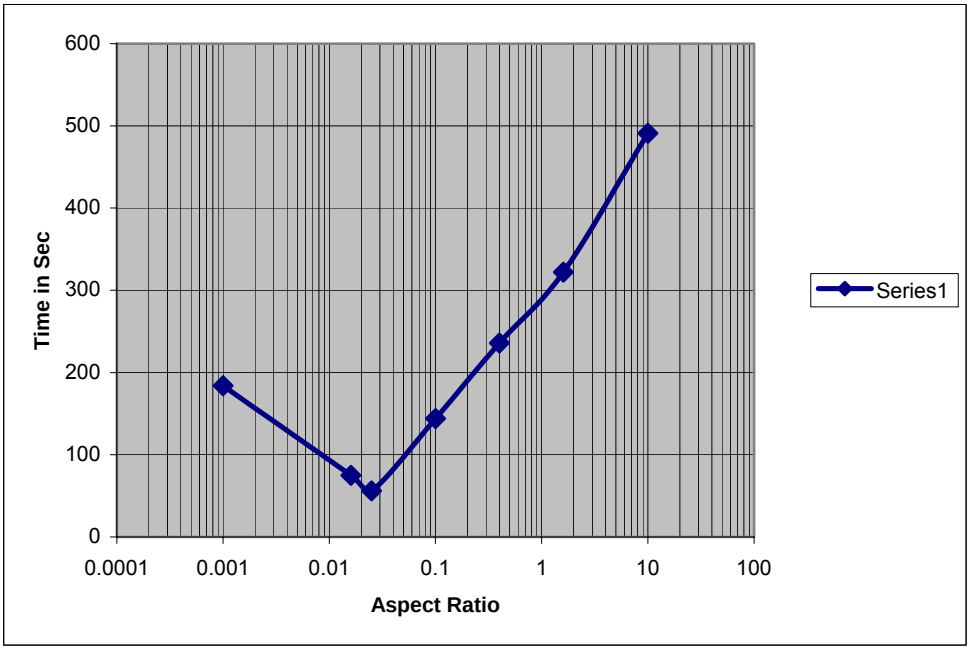


Figure 24: Influence of aspect ratio on computation speed using ECLIPSE 100 (1-D domain decomposition).

Case 2) Aspect ratio constant while the number of grid blocks were changed

The same reservoir scenario was simulated but this time with different grid resolution and keeping the aspect ratio constant. The grids employed are summarized below:

Grids	No. of grids	Time (Sec)
4*4*3	48	4
40*40*3	4800	21
80*80*3	19200	76
160*160*3	76800	530

The results shown in Figure 25 indicate that maximum gain in computational speed is achieved when the grid size is increased by a factor of 100 from 100 grid blocks to 10000 blocks. Subsequently, the computation slows down when the grid size is increased. This could be attributed to the increased communication between processors due the larger interacting surface area among the domains.

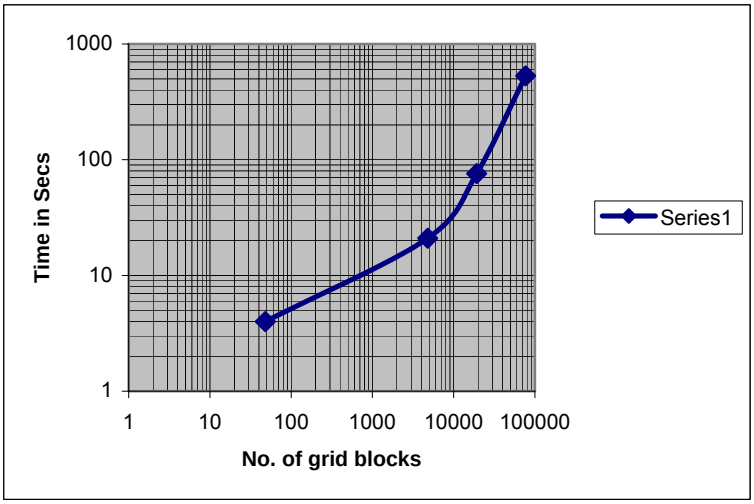


Figure 25: Plot of computation time against grid size. Non-linear scaling of computation speed with increase in grid size is indicated.

Case 3) Number of grid blocks are kept same while the aspect ratio and the physical dimensions of reservoirs are varied

The reservoir is assumed to have constant permeability of 500 md, constant porosity of 30%. The reservoir is assumed to have one vertical producer in the middle of the reservoir. Four parallel processors were utilized for the parallel simulation. Parallel Eclipse 100 was used that allows the reservoir to be divided in either X or Y dimension. Three cases were run, with identical reservoir dimensions and well locations. The simulation was run for 500 days. Parallel simulations utilized four processors.

Physical dimension of grid blocks were kept same i.e. 500 ft, 500 ft and 50 ft in x, y and z directions respectively.

Aspect Ratio	Time in sec
0.001	598
0.025	213
0.1	100
10	82
1000	413

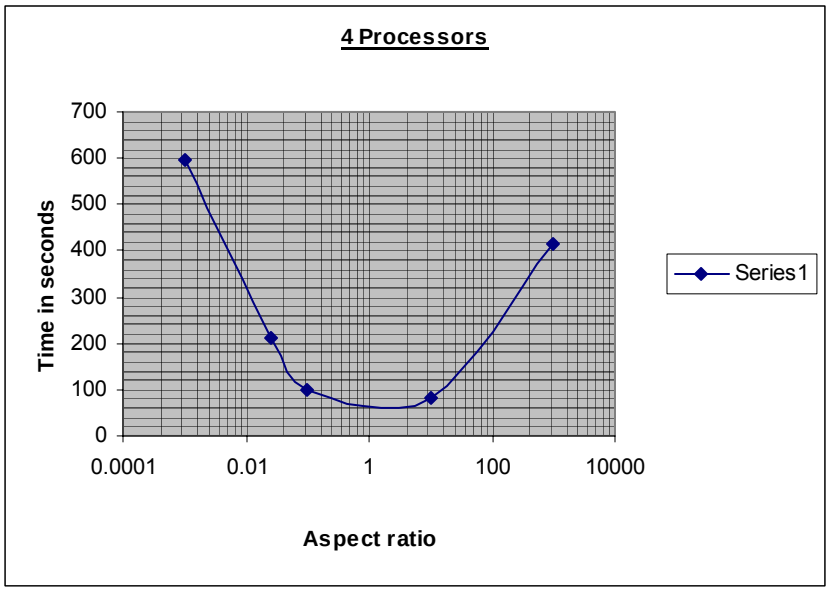


Figure 26: Plot of computation time versus aspect ratio while adjusting the grid block sizes so as to keep the reservoir dimension the same.

The results indicate that simulations performed with approximately the same number of blocks in the x and y directions and with square grid blocks are the most efficient.

Flux Boundary Conditions

The following are the results obtained while simulating a small section of the field using the boundary conditions obtained from the full-scale simulation. The full field has 50*50 gridblocks while the smaller sections have 25*25 grid blocks. Fluxes (flows of each phase from the full field run as a function of time) are used as boundary conditions. The full field simulation took 21 seconds while the smaller section took 9 seconds. Figure 27 shows the pressures and oil saturations at the end of 5000 days. Comparing the grid bloc pressures and saturations obtained for the sub domain simulations against the full field

simulation values, the accuracy of the sub-domain simulation results was ± 1 psi. The saturations were alike up to three places of decimal.

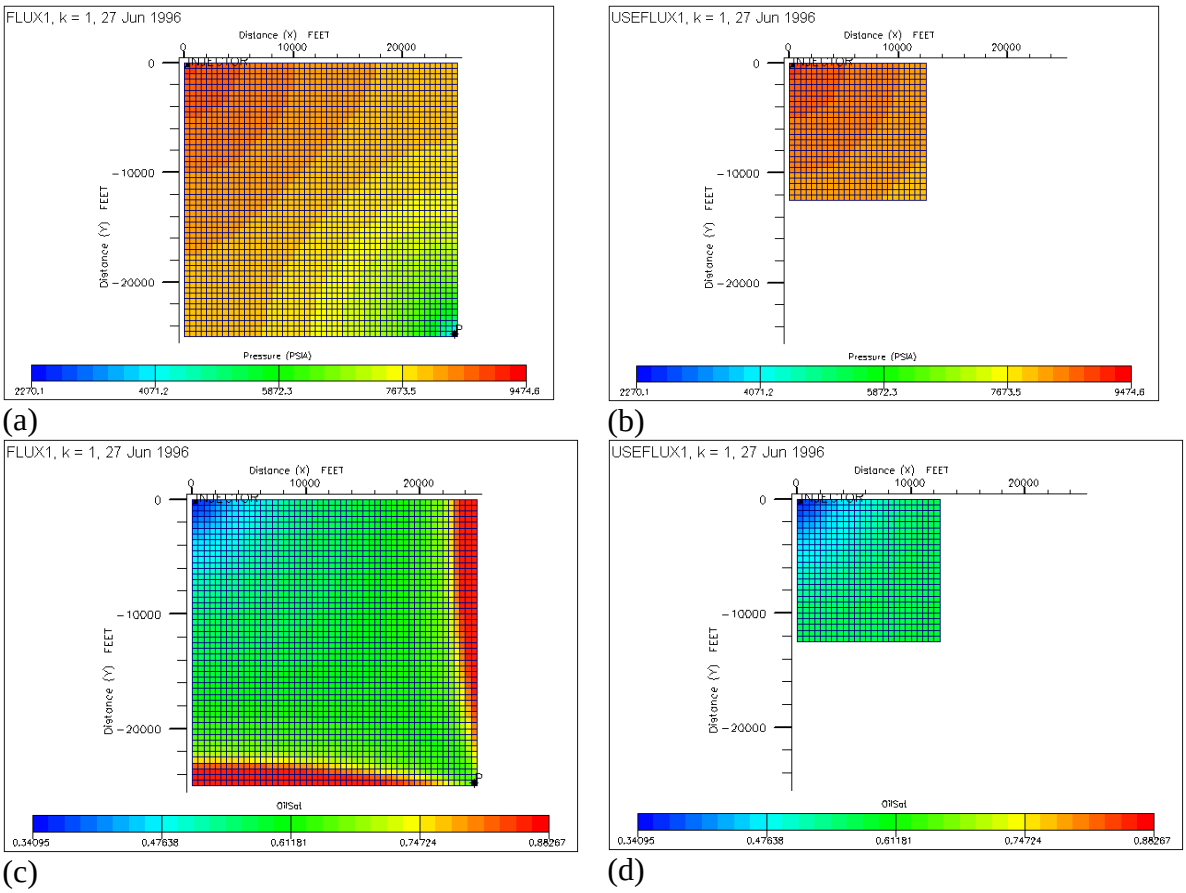


Figure 27: Full field flow simulation results: a) Pressure variations in the reservoir at 500 days obtained by full field simulation; b) Pressure variations in a domain obtained corresponding to boundary conditions derived from the full field simulation; c) Oil saturation values obtained by full field simulation, and d) Oil saturation values obtained for sub-domain simulation using boundary conditions derived from full field simulation.

The choice of simulator was dictated by the availability of the calculation options that facilitate identification of sensitive regions and provide Hessian matrix of the objective function. Various Parallel and Boundary flux cases were run for the preliminary checks on Eclipse. A scheme for using parallel and distributed computing, specifically from the project's point of view has been formulated.

Putting Things Together – A Gradual Updating Procedure for History Matching

An approach that uses a probability perturbation method for gradual deformation of geological models conditioned to dynamic information has been presented. This approach, compared to other perturbation methods, offers the important advantages of preserving the prior geological heterogeneity model and simplifying the history match process to a single (or few) parameter(s) optimization problem.

The reproduction of historical production data is a complex non-linear inverse problem. This implies that the probability updating cannot be accomplished in a single perturbation loop starting from an initial realization of the permeability field; calibrates an optimal r_D value and obtains an updated probability distribution reflecting the dynamic characteristics of the reservoir. Instead, a multi-loop iterative process is required to update the geological model using the dynamic data.

A Markov-Chain is a stochastic updating procedure where the parameter state at any step of the procedure is assumed to be dependent only on the state immediately prior to that step. Thus the proposed realization at any stage of the process depends only on the preceding realization in the sequence, and the convergence towards the desired realization depends on carefully specifying the transition from one realization to the next one, i.e. the methodology for the new proposed realization. In this case, the parameter r_D controls the transition of the permeability value at a location from one category to the next. The calibration of the deformation parameter to condition the gradual deformation

of the geological model to dynamic information represents the internal optimization scheme. The converged model and realization at the end of the inner loop is then used as the starting realization for the next sequence of inner Dekker-Brent optimization runs to determine the conditional distribution P(A|C).

In the implemented Markov chain approach, on every updating step, from iteration step l to step $l+1$ of the outer loop, the probability distributions conditioned to dynamic and static information, $P(A|B,C)^{l+1}$, are obtained by applying the permanence of ratio hypothesis to combine distributions conditioned to static information. The distribution $P(A|B)^{l}$ is obtained from geological data and heterogeneity model and the distribution conditioned to dynamic information, $P(A|C)^{l+1}$ is estimated knowing the indicator category at each location from the realization sampled from $P(A|B,C)^{\ell}$, the prior distribution P(A) and the deformation parameter, r_D, calibrated using the Dekker-Brent iterative optimization procedure. The converged distribution $P(A|B,C)^{l+1}$ is then used as the starting point for the next inner Dekker-Brent loop until global match to the historic data is attained.

Even though the two-loop Markov chain procedure ensures global convergence, the introduction of multiple sets of inner optimization schemes requires multiple evaluations of the flow response. However, the dynamic-calibrated gradual deformation methodology renders the history match process faster and more controlled, increasing the consistency

between the initial and the proposed realizations at every step, and improving the rate of convergence of the objective function.

Preliminary evaluations of the algorithm for dynamic data integration using the proposed probabilistic approach have been pursued with the synthetic case described in the following table.

TABLE 1. Description Of The Synthetic Simulation Case Used To Evaluate The Probabilistic Dynamic Data Integration Algorithm.

SIMULATION PROPERTY/DESCRIPTION	VALUE
Simulation Model	Black Oil
Solution	Implicit
Simulation Period, years	2
Grid (Cartesian)	50x50x5
Active Grid blocks	12500
Grid block dimensions, ft^3	80x80x4
Porosity	0.22
Kx = Ky (Mean – Std Dev), md	200 - 250
Kz/Kx	0.15
Saturation Pressure, psi	5064
Water-Oil Contact, ft	9000
Gas-Oil Contact, ft	4000
Reference Depth, ft	7300
Initial Pressure @ 7300 ft, psi	6000
Residual Water Saturation	0.18
Residual Oil Saturation	0.24
Oil Relative Permeability Endpoint	0.7
Water Relative Permeability Endpoint	0.5
Oil Gravity (API)	35
Water Injectors	1
Injection – Control Rate, Stb/day	5000
Injection – BHP upper Limit, psi	8000
Oil Producers	2
Production – Control BHP Lower limit, psi	2000
Production – Minimum rate, Stb/day	10

Figures 28-32 describe the convergence characteristics of the algorithm on the synthetic case study.

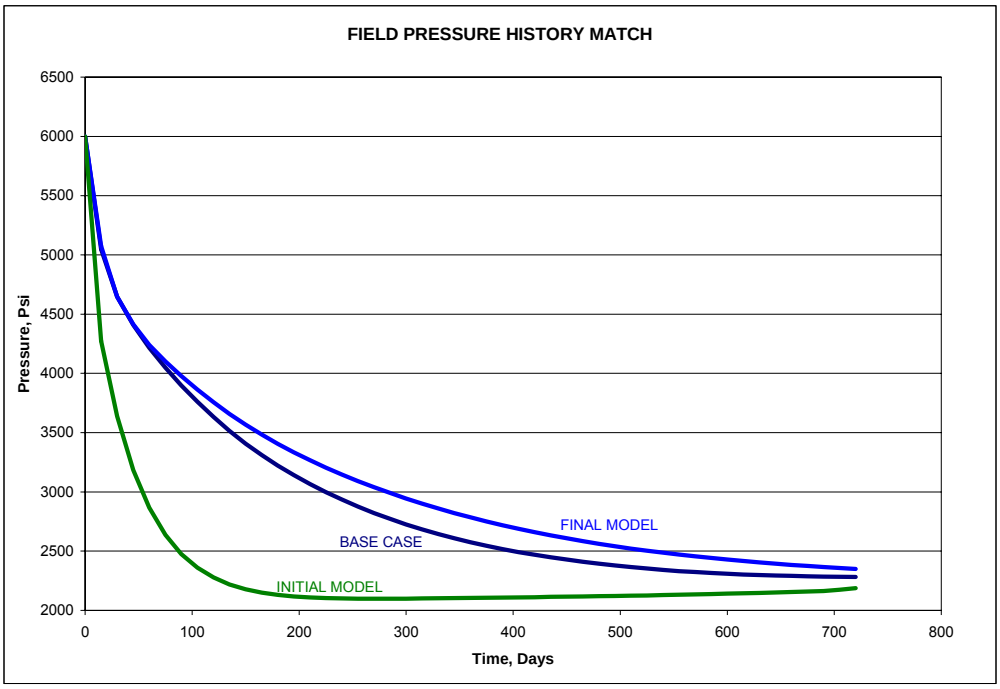


Figure 28: Field Pressure History Match obtained with the probabilistic dynamic data integration algorithm for the synthetic case study. Results after forty flow simulation runs.

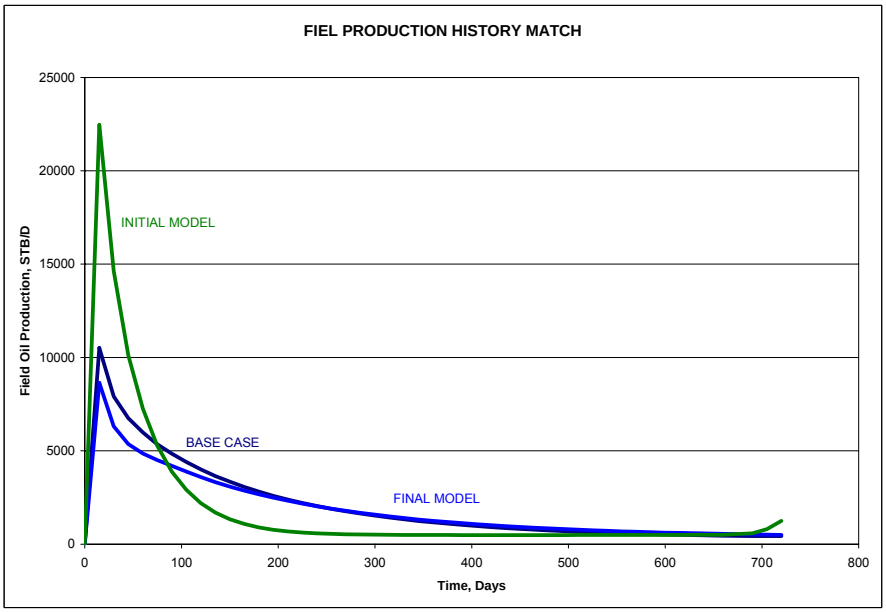


Figure 29: Field Production History Match of synthetic case using the probability perturbation method. Final model is obtained after 40 simulation runs.

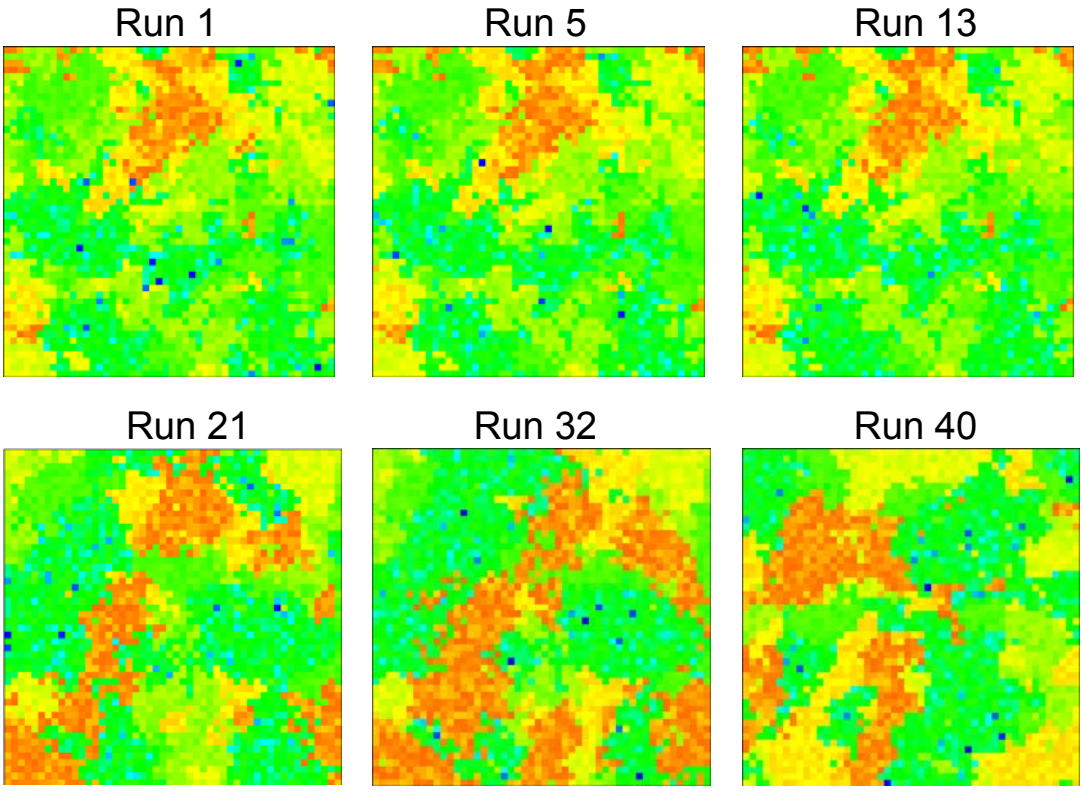


Figure 30: Example of Gradual Deformation of geological models, obtained with the probability perturbation method. Variations in the Third layer of the Geological model through 40 flow simulation runs are presented.

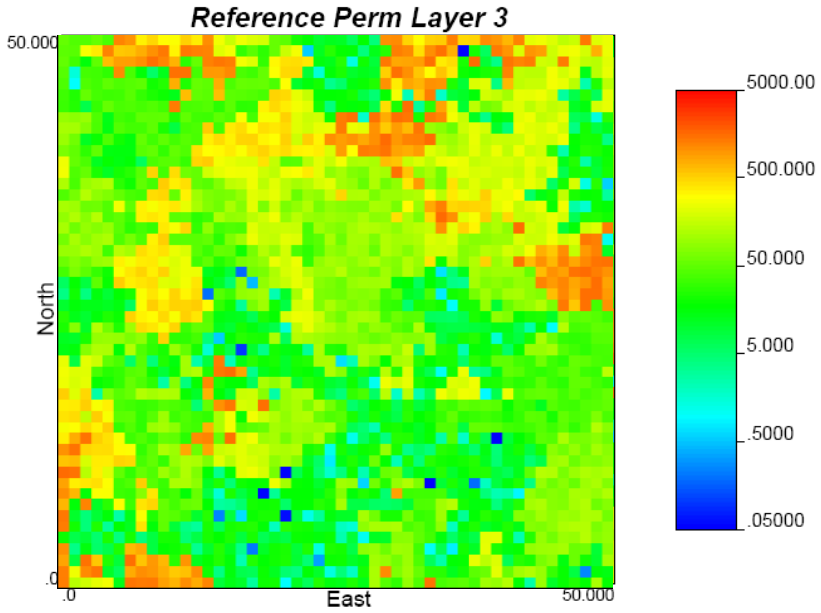


Figure 31: Third layer of the reference geological model used in the synthetic case to generate the historical production data.

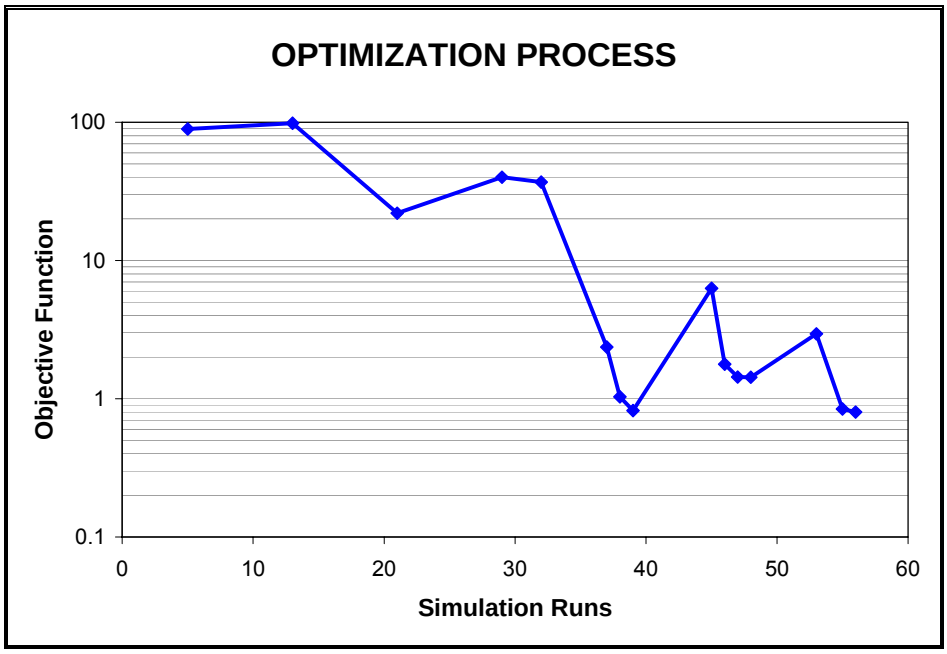


Figure 32: Example of Convergence in the objective function applying the probabilistic approach for assisted history matching by gradual updating of conditional distributions using dynamic information.

CONCLUSION

The general procedure for gradual updating of geological models within an assisted history matching framework can be summarized in the following steps:

1. An indicator sequential simulator is used to generate an initial stochastic realization of the target reservoir model and calculate the local probability distributions conditioned to static information.
2. A Markov chain iterative updating process is started with the initial realization. The Markov chain forms the outer loop of the procedure and every outer step or outer iteration includes the following sub procedures:

2.1. Generate a new set of random draws to sample from the local conditional distributions. This set of sampling draw is fixed during each outer iteration, but changes from an outer iteration to another.

2.2. Evaluate the Objective function at different values of the deformation parameter, rd, screening the whole range of variability [0, 1]. Usually 5 different values are enough to get started. For each value of rd, a different geological model is obtained and run in the flow simulator; and the objective function is evaluated.

2.3. Pick the value of the deformation parameter with the minimum objective function and start the calibration process of rd with the dynamic data using the Dekker-Brent iterative algorithm. This calibration process is called the inner loop, and the number of inner steps or inner iterations can be fixed or controlled by a tolerance in the change of the objective function in consecutive steps.

2.4. Use the best model (with the minimum objective function) to update the stochastic realization. When the best model is obtained with a deformation parameter of zero, no updating is required (the realization remains invariant). This is the final step of the outer loop.

3. Repeat step 2 (outer loop) until a tolerance in the objective function (history match) has been reached or for a fix number of outer iterations.

4. Print out final geological realization with the corresponding flow response.

Analysis of schemes for optimal parameterization of the history matching procedure in a parallel and distributed computing environment leads to the following conclusions:

1) Global rather than well specific objective function should be minimized by perturbing permeability values locally within domains.

2) Reservoir regions based on sensitivity coefficients calculated for an initial guess of the permeability field that is far from the “truth” may not be very reliable. It is therefore necessary to re-compute the reservoir regions as the permeability field is being updated during the history matching process.

3) The grouping of correlated grid blocks (from flow perspective) has been proposed by some researchers as a way to simplify the history-matching problem by decreasing the number of parameters to be optimized. In this project, the feasibility of using the same concept to define reservoir regions has been explored. The procedure ensures that the defined domains are least correlated in terms of their effect on the objective function.

4) Domains have been defined by performing principal component analysis of the Hessian matrix (sensitivity coefficients) and the permeability covariance matrix and then subsequently applying a cut-off threshold in order to obtain reservoir regions that span a reasonable volume of the reservoir. A variance threshold may also be specified in order to reduce the number of eigen vectors (regions).

5) Domains have also been defined by first suitably scaling the eigenvectors then applying a single threshold rather than different threshold for different eigenvectors. Further testing of this concept for reducing sensitivity to user specified parameters are required.

6) In case there is high degree of overlap among the eigen-components of different eigenvectors, rotation of eigenvectors using a variance maximization procedure is proposed and has been tested.

7) While doing Principal Component analysis of permeability coefficient matrix, regions with the highest uncertainty (variance) are identified as the most sensitive. However, a drawback with using static data for domain delineation is that it is very much possible that there is very high flow across facies due to particular well

locations i.e. facies are correlated from flow perspective but not statically. Also high variance (or high sensitivity statically) regions may have not much flow associated and thus would not affect the objective function in any significant way.

8) Domain decomposition based on porosity and thickness was tried. From the test cases it appeared that sensitivity coefficients are a strong function of the permeability realization in consideration. Geobodies defined on the basis on sensitivities and coefficients were inadequate to define the domains.

9) In order to counter the high computational cost associated with sensitivity analysis an approximation based on analysis of limited duration of production data has been proposed. The delineated zones were determined to be robust.

10) A scheme for history matching using the domains in the distributed computing environment has been discussed.

REFERENCES

1) Richard Reyment, K.G. Joreskog, “Applied Factor Analysis in the Natural Sciences”

2) Bissell, R. “Calculating Optimal Parameters for History Matching”: 4th European Conference on the Mathematics of Oil Recovery, Roros, Norway

3) R.C. Bissel, Yogeshwar Sharma, J.E. Killough, “History Matching Using the Method of Gradients: Two Case Studies”, SPE 28590

4) Richard L. Gorsuch, “Factor Analysis”, Second Edition.

5) Leitao, H.C.; Schiozer, D.J., “A new automated History Matching Algorithm improved by parallel computing”, SPE 53997, SPE Latin American and Carribbean Petroleum Engineering Conference held in Venezuela, 1999.

6) W.A. Habiballah, M.E. Hayder, M.S. Khan, K.M. Issa, S.H. Zahrani, R.A. Shaikh, A.H. Uwaiyedh, T.P. Tyraskis, M.A. Baddourah. “Parallel reservoir simulation Utilizing PC Clusters”, SPE 84065, SPE Annual Technical Conference held in Colorado, 2003.

7) D.J. Schiozer, S.H.G. Sousa, “Use of Reservoir simulation, Parallel computing and Optimization techniques to accelerate History Matching and Reservoir management decisions”, SPE 53979, Fifth Latin American and Caribbean Petroleum Engineering Conference and Exhibition, Brazil, 1997.

8) D.J. Schiozer, S.H.G. Sousa, “Use of external parallelization to improve history matching”, SPE 39062, Fifth Latin American and Carribean Petroleum Engineering Conference and Exhibition held in Rio De Janeiro, 1997.

9) Ahmed Quenes, William Weisss, A.J. Sultan, Abu Dhabi Nati, Jaber Anwar, “Parallel Reservoir Automatic History matching using a network of Workstations and PVM”, SPE 29107, SPE Symposium on Reservoir Simulation held in San Antonio, 1995.

10) Mengjiao Yu, Phani B. Gadde and Mukul M. Sharma, “Process based Decomposition on Distributed Systems”, SPE 77721, Annual Technical Conference and Exhibition held in San Antonio, 2002.

11) J.E. Killough, “Is parallel computing ready for reservoir simulation”, SPE 26634, 68th Annual Technical Conference and Exhibition of Petroleum Engineers held In Houston, 1993.

12) Ahmed Quenes, Naji Saad, “A New Fast Parallel Simulated Annealing Algorithm for Reservoir Characterization”, SPE 26419, 68th Annual Technical Conference and

Exhibition of SPE held in Houston, 1993.

13) Luc Anguille, J.E. Killough, T.M.C. Li and J.L. Toepfer, “Static and Dynamic Load Balancing Strategies for Parallel Reservoir Simulations”, SPE 29102, 13th SPE Symposium on Reservoir Simulation held in San Antonio, 1995.

14) Jorge L. Landa, Sebastien Strebelle, “Sensitivity analysis of Petrophysical Properties Spatial Distributions and Flow Performance Forecasts to Geostatistical Parameters using Derivative Coefficients”, SPE 77430, SPE Annual Technical conference in San Antanio, 2002

15) H.A. Tchelepi, L.J. Durlofsky, W.H. Chen, A. Bernath and M.C.H. Chien, “Practical use of scale up and Parallel Reservoir Simulation Technologies In Field studies”, SPE 38886, SPE Annual Technical Conference and Exhibition in San Antonio, 1997.

16) C.E. Romero, J.N Carter, R.W. Zimmerman, A.C. Gringarten, “Improved Reservoir Characterization through Evolutionary Computation”, SPE 62942, SPE Annual Technical Conference and Exhibition in Dallas, 2000.

17) F. Roggero, L.Y. Hu, “Gradual deformation of Continuous Geostatistical Models for History Matching”, SPE 49004, SPE Annual Technical conference in Louisiana, 1998.

18) Bertrand Brun, Olivier Gosselin, John W. Barker, “Use of prior information in Gradient-Based History Matching”, SPE 66353, SPE Resevoir Simulation Symposium held in Houston, 2001.

19) Suasana Gomez, Olivier Gosselin, John W. Barker, “Gradient Based History Matching with Global Optimization Method”, SPE 56756, Annual Technical Conference of SPE held in Houston, 1999.

20) P.C. Shah, G.R. Gavalas, J.H. Seinfeld, “Error analysis in history matching: The optimum level of Parameterization”, SPE 6508, 1977.

21) Sophie Verdiere, Lisette Quettier, Pierre Samier, Alan Thombson, “Application of Parallel simulator to industrial cases”, SPE 51887, SPE Reservoir Simulation Symposium held in Houston, 1999.

22) Eclipse Technical Manual, 2003A, Schlumberger.

23) Iterative Updating of Reservoir Models Constrained to Dynamic Data. CIM paper 2002-125, Kashib, T. and Srinivasan, S

24) Ahmed Quenes, B. Brefort, Gilbert Meunier, Simon Dupere, “A New Algorithm for Automatic History Matching: Application of Simulated Annealing Method to

Reservoir Inverse Modeling, SPE 26297.

25) Schiozer, D.J., "Use of Reservoir Simulation, Parallel Computing and Optimization Techniques to accelerate History Matching and Reservoir Management Decisions", SPE 53979, SPE Latin American and Carribean Petroleum Engineering Conference held in Venezuela, 1999.

26) Blanc, G., Hu, L.Y., Noetinger, B., "Gradual deformation and iterative cal liberation of sequential stochastic simulation", Mathematical Geology, V.33, N.4:475-489, 2001.

27) Caers, J: "Markov chain theory for spatial stochastic simulation". Report No. 12, Volume 1, May 1999, Stanford for Reservoir Forecasting, Stanford University.

28) Caers, J: "Methods for history matching under geological constrains", 8th European Conference on Mathematics of Oil Recovery, 2002.

29) Chu, L., Renyolds, A.C. and Oliver, D.S.: " Computation of sensitivity coefficients for conditioning the permeability field to well test pressure data", In Situ, V.19, N.2:179-223, 1995.

30) J.E. Killough, M.F. Weeler, "Parallel Iterative Linear Equation Solvers: An investigation of Domain Decomposition Algorithms for Reservoir Simulation", SPE 16021, Ninth SPE symposium on reservoir simulation held in San Antonio, 1987.

31) Sanjay Srinivasan, Steven Bryant, "Integrating Dynamic Data in Reservoir Models Using a Parallel Computational Approach", SPE 89444

32) R.W. Schulze-Riegert, O. Haase, A. Nekrassov, "Combined Global and Local Optimization Techniques Applied to history Matching", SPE 79668

33) Jef Caers, "Methods for History Matching under Geological Constraints"

34) Th. Grussaute, P. Gouel, " Computer Aided History Matching of a Real Field Case", SPE 50642

35) R.C. Bissel, O. Dubrule, P. Lamy, P. Swaby and O. Lepine, " Combing Geostatistical Modelling With Gradient Information for History Matching: The Pilot point Method", SPE 38730

36) Ning Liu, Dean S. Oliver, "Automatic History Matching of Geological Facies" SPE 84594

37) Eliana L. Ligero, Celio Mashio, Denis J. Schiozer, “Quantifying the Impact of Grid Size, Upscaling, and Streamline Simulation in the Risk Analysis Applied to Petroleum Field Development”’ SPE 79677

LIST OF ACRONYMS AND ABBREVIATIONS

Measurement Units – All reports to be delivered under this instrument shall use the SI Metric System of Units as the primary units of measure. When reporting units in all reports, primary SI units shall be followed by their U.S. Customary Equivalents in parentheses ().

The Recipient shall insert the text of this clause, including this paragraph, in all subcontracts under this award.

Note: SI is an abbreviation for “le Systeme International d’Unites.”

APPENDICES

None.